Fundamental Limits of Learning for Generalizability, Data Resilience, and Resource Efficiency

by

Moïse Blanchard

M.S. in Applied Mathematics, École Polytechnique, 2019

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Moïse Blanchard
Operations Research Center
May 3, 2024

Certified by: Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center
Thesis Supervisor

Accepted by: Georgia Perakis
John C Head III Dean (Interim), MIT Sloan School of Management
Professor, Operations Management, Operations Research & Statistics
Co-Director, Operations Research Center

Fundamental Limits of Learning for Generalizability, Data Resilience, and Resource Efficiency

by

Moïse Blanchard

Submitted to the Sloan School of Management
on May 3, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

ABSTRACT

With the advancement of machine learning models and the rapid increase in their range of applications, learning algorithms should not only have the capacity to learn complex tasks, but also be resilient to imperfect data, all while being resource efficient. This thesis explores trade-offs between these three core challenges in statistical learning theory. We aim to understand the limits of learning algorithms across a wide range of machine learning and optimization settings, with the goal of providing adaptable, robust, and efficient learning algorithms for decision-making.

In Part I of this thesis, we study the limits of learning with respect to generalizability and data assumptions following the universal learning framework. In universal learning, we seek general algorithms that have convergence guarantees for any objective task without structural restrictions. While this cannot be achieved without conditions on the training data, we show that in general this can be performed beyond standard statistical assumptions. More generally, we aim to characterize provably-minimal assumptions for which universal learning can be performed, and to provide algorithms that learn under these minimal assumptions. After giving a detailed overview of the framework and a summary of our results in Chapter 2, we investigate universal learnability across a wide range of machine learning settings: full-feedback in realizable online learning (Chapter 3), supervised learning with arbitrary or adversarial noise (Chapter 4); partial-feedback in standard contextual bandits (Chapter 5) and, as a first step towards more complex reinforcement learning settings, contextual bandits with non-stationary or adversarial rewards (Chapter 6).

We investigate the impact of resource constraints in Part II, specifically of memory constraints in convex optimization. The efficiency of optimization algorithms is typically measured through the number of calls to a first-order oracle which provides value and gradient information on the function, aptly referred to as oracle-complexity. However, this may not be the only bottleneck; understanding the trade-offs with the usage of resources such as memory could pave the way for more practical optimization algorithms. Following this reasoning, we make advancements in characterizing achievable regions for optimization algorithms in the oracle-complexity/memory landscape. In Chapter 7 we show that full memory is necessary to achieve the optimal oracle-complexity for deterministic algorithms; hence, classical cutting-plane methods are Pareto-optimal in the oracle-complexity/memory trade-off. On the positive side, we provide memory-efficient algorithms in Chapter 8 for high-accuracy

regimes (sub-polynomial in the dimension). In exponential-accuracy regimes, these algorithms strictly improve the oracle-complexity of gradient descent while preserving the same optimal memory usage. These algorithms can in fact be used for the more general feasibility problem for which we give improved lower-bound trade-offs in Chapter 9. These results imply that in standard accuracy regimes (polynomial in the dimension), gradient descent is also Pareto-optimal and reveal a phase transition for the oracle-complexity of memory-constrained algorithms.

Thesis supervisor: Patrick Jaillet
Title: Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Acknowledgments

I am deeply grateful to my advisor Patrick Jaillet, whose unwavering support and gentle guidance have shaped my PhD journey through these years. Patrick taught me the values of academic rigor and intellectual curiosity, and gave me invaluable insights into the academic environment. I am indebted to him for the amazing freedom he granted me while I was exploring research directions, even when these were quite uncertain in the first few years of my PhD. Patrick was always extremely patient and supportive throughout this journey, helping me grow personally and intellectually, and I hope that I can emulate his great mentorship skills in my future endeavors.

I would also like to thank David Gamarnik and Alexander Rakhlin for offering great advice and valuable feedback on my thesis. I hold great admiration for them as scholars, but also as teachers and mentors, and am honored that they agreed to join my thesis committee.

I also wanted to extend my warm gratitude to Steve Hanneke, who inspired most of my thesis. Steve is an exceptional researcher with an amazing vision; his capacity to ask elegant and fundamental learning questions is incredible. I was so fortunate to start my learning theory journey through one of his questions on universal learning. When I first met Steve at a conference in Boulder, I had no exposure to learning theory, yet upon seeing the deep and beautiful questions he asked, I was immediately hooked. I also had the chance to collaborate with him and learned so much from his passion for research.

I want to express my sincere gratitude to Nathan Srebro and Blake Woodworth who have also greatly influenced my academic journey. After working on one of their questions, Nathan showed me amazing support and encouragement. I am grateful that he invited me to visit his amazing research group at Toyota Technological Institute in Chicago; I learned so much from our research discussions.

I am very grateful to Aryeh Kontorovich for offering constant support throughout my PhD. I was also very fortunate to collaborate with him towards the end of my PhD journey, and was inspired by his dedication to his students and his research.

I am extremely grateful to all the people who mentored me throughout these years. In particular, I want to thank Alexandre Jacquillat for believing in me when I started my PhD. I had the chance to work with him at the beginning of my PhD, and Alexandre was always very thoughtful and supportive. He introduced me to different types of research, and I also learned from his amazing writing skills. I am very grateful to Gregory Valiant for his kind words of encouragement. I learned a lot from reading works from him and his research group; these served as a strong stepping stone for the second part of this thesis. I also wanted to thank Ryan Cory-Wright, Thodoris Lykouris, Swati Gupta, Bart Van Parys, Mark Sellke, Tianyi Peng, and Jackie Baek for giving detailed advice throughout my academic job search.

I want to thank Jesús De Loera. I was honored to spend a summer doing research with him before the PhD. He is one of the most captivating and passionate researchers I have met, and he is part of the reason I decided to do a PhD in the first place. Last, thank you 廖老师 for teaching me and putting up with my poor Chinese.

I can't thank all of my friends enough. Their support was crucial throughout the adventures of the PhD. I wanted to thank my dear friend Shuvomoy Das Gupta, I am deeply inspired by his life trajectory, his strong values, perseverance, wisdom, and humility. He was a great mentor for me; I am so grateful to count him among my closest friends and I cannot wait to see his next endeavors. I am extremely thankful to Romain Cosson who has been a very close friend since undergrad. I have always been encouraged by his self-driven motivation and optimism. When he joined MIT, he was the first to encourage me to study learning theory topics, which turned out to have a profound impact on my future research trajectory. Collaborating with Romain was also one of the most fun times of my PhD.

I was extremely fortunate to be part of a close group of caring friends at MIT that I shared so many dear moments with. In particular, I warmly thank Amine Benounna, Léonard Boussioux, Ben Lahner, Athul Paul Jacob, Yu Ma, Vassilis Digalakis, Liviu Aolaritei, Tanay Wakhare, Kimberly Villalobos Carballo, Cynthia Zeng, Arnaud Robin, Baptiste Rabecq, Baptiste Rossi, Gabriel Afriat, Zikai Xiong, Kamessi Zhao, Rares Christian, Brian Liu; and my friends and collaborators Adam Quinn Jaffe, Junhui Zhang, and Václav Voráček.

I am deeply grateful to my close friends from undergrad and high school, Matthieu, Samy, Christian, Cécile, Pierre, Antoine, Sophie, Inès, Randy, Thomas, Cyril, David, Julien, Antonin, and Guillaume. We always have incredibly fun times together.

I am especially grateful to you, Emily, for illuminating my life during the PhD. I could never have hoped for such an amazing partner, friend, and confidante. The moments and memories that we shared are the highlights of my PhD journey, and I so very much look forward to sharing many more years to come.

Finally, I want to extend my gratitude to my family, to my brother Benjamin and sister Aurélie, and most importantly to my parents, Victoria and Gabriel Blanchard, without whom none of this would have been possible. They sacrificed a lot so that we could have the best possible life and showed us unconditional love and support throughout all these years. This thesis is dedicated to them.

The work in this thesis was partly supported by ONR grant N00014-18-1-2122 and AFOSR grant FA9550-23-1-0182.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Motivation

With the rapid advancement of machine learning (ML) and the increasing reliance on automated decision-making, a major challenge in statistical learning is to design learning algorithms that are not only efficient but also enjoy performance guarantees for large classes of problem instances. This is especially relevant as ML tools are being used in high-stakes domains where errors can have serious consequences. Additionally, the task to be learned may be very complex, and the data used to train these algorithms may be unstructured or even adversarially corrupted. In this thesis, we touch upon the following three major challenges.

Generalizability. To deploy a machine learning model for new applications, we need to ensure that it can learn large classes of tasks. As machine learning has evolved, data-driven models have shown that they can handle increasingly more complex problems, reaching amazing levels of performance in tasks such as generating images, videos, or language. To learn such complex tasks, choosing a good learning model is a crucial and challenging step. On one hand, more complex learning models—often characterized by a larger number of training parameters—can represent more complex decision functions. However, classical issues such as overfitting may also need to be taken into account.

The standard approach in statistical learning theory is to first choose a set of functions, also called a *function class*. We then aim to achieve performance comparable to the best function within this class for the specific problem at hand. For the aforementioned ML models, a natural choice of function class is simply the class of functions that can be represented by the model. As a concrete example, in linear regression one aims to learn the best fit within the function class of linear predictors. Of course, having guarantees for larger function classes is more desirable since the algorithm can then tackle more *general* classes of problems.

Data resilience. Real-world data is often far from ideal; available data can be highly noisy, unstructured, correlated, or even adversarially corrupted. To provide theoretical guarantees on learning algorithms, assumptions on the data generating process are however necessary. For instance, algorithms may require the observed data to be representative of, or reasonably

cover the class of instances on which it is tested: we would not expect an algorithm to perform well on instances for which it never observed any similar data. As another simple example, if the data contains too many adversarial corruptions, the task of recovering the true patterns in the underlying problem may be impossible.

The most classical data assumption from far is that each available example was generated from independent and identically distributed (i.i.d.) processes. This assumption has been instrumental in achieving learning guarantees, in fact the analysis for the i.i.d. case sometimes can be lifted to relaxations of the i.i.d. assumption for data processes, such as processes with mixing, stationary, or ergodic properties. Going beyond the i.i.d. data assumptions and learning in adversarial environments has hence become an important question in statistical learning, under the umbrella terms of data *robustness* or data *resilience*.

Resource efficiency. Large-scale automated decision-making faces various bottlenecks, including standard computational considerations such as runtime, memory usage, and energy consumption, as well as societal and ethical concerns, among others. While the traditional measure of algorithms' efficiency largely remains runtime or computational complexity, understanding the impact of the other practical constraints has become increasingly important in recent research. In this thesis, memory usage will be a key focus area, given its significant role in large-scale optimization, in particular during model training.

The overarching goal of this thesis is to explore the fundamental limits of learning along these objectives in optimization and statistical learning. A recurrent theme will be to understand the trade-offs between these challenges, which in turn guides the design of algorithms. We divide this thesis into two main parts, which are briefly summarized below. Within Part I, Chapter 3 is based on the works [Bla22; BCH22; BC22], Chapter 4 is based on [BJ23], Chapter 5 is based on [BHJ22], and Chapter 6 is based on [BHJ23]. Within Part II, Chapter 7 is based on [BZJ23], Chapter 8 is based on [BZJ24], and Chapter 9 is based on [Bla24].

1.2 Universal Learning

In Part I, we study the limits of learning with respect to both generalizability and data assumptions. This framework was introduced by [Han21a] to study learning under minimal assumptions. A detailed overview of the motivation, framework, and results is given in Chapter 2.

It is well understood that positive guarantees for learning cannot be achieved both without restrictions on generalizability (arbitrarily general target tasks) and without data assumptions (adversarial data). Previous works exhibited a fundamental trade-off between these two objectives. On one extreme, learning with adversarial data can only be performed for restricted function classes: typically those that have finite so-called Littlestone dimension [Lit88]. In particular, this does not give any positive results even if the function class only contains linear functions. Instead, one can make assumptions on the data-generating process to account for richer benchmark function classes. Among many other important results, it

is known that under i.i.d. data, one can efficiently learn finite VC dimension function classes [VC71], which encompasses linear function classes.

Universal learning lies at the other extreme and aims to provide learning guarantees without any prior assumption on the form of the target task. The goal is therefore to have algorithms that are fully general and can learn any arbitrary target decision function, arbitrarily complex though it may be. While this of course a very strong objective, achieving *consistency* results is still possible: the decisions of the algorithm converge to the optimal decisions. In fact, early works already showed that for i.i.d. data and say binary classification, such universal consistency results can be achieved with very simple nearest-neighbor algorithms [CH67; Sto77; DGL13]. Subsequent works also succeeded in generalizing these results for stationary ergodic processes [Orn78; Alg92; MYG96; GL02; Nob03]. Our goal is to understand the *provably-minimal* assumptions on the data-generating processes for which this learning guarantee can still be achieved.

1.2.1 Universal Realizable Online Learning

In Chapter 3, we focus on the simplest model for sequential prediction tasks, in which a learner iteratively observes a context $X_t \in \mathcal{X}$ containing any form of information about the new instance, makes a prediction $\hat{Y}_t$ about the corresponding value, and finally observes the true value $Y_t \in \mathcal{Y}$ which can be used to update its future predictions. In the *realizable* setting, we assume that the responses do not have any noise, that is, there is some prediction function $f^\star$ that perfectly describes the responses $Y_t = f^\star(X_t)$. In this context, the goal is to converge to the optimal predictions, which is measured in terms of average loss between predictions $\hat{Y}_t$ and true values Y_t. Such predictions are said to be consistent, and an algorithm is *universally consistent* if its predictions are consistent for any underlying optimal prediction function $f^\star$. This simplified model will be our starting point for studying universal learning, in which, no assumptions are made on the specific learning target $f^\star$. As mentioned above, prior works showed that for bounded losses, universal consistency was possible in this setting for i.i.d. context sequences $(X_t)_{t\geq 1}$ or even stationary ergodic.

To understand the minimal data assumptions, we first provide a necessary and sufficient characterization of the class of processes for which universal learning is possible. For the main case of bounded losses, we refer to this class of processes as Sublinear Measurable Visits (SMV), which is very general and significantly generalizes stationary ergodic processes and other classical statistical assumptions. It intuitively asks that the process $(X_t)_{t\geq 1}$ does not explore too many different regions of the instance space. Second, we provide algorithms that can ensure universal consistency under these provably-minimal data assumptions: these are called *optimistically universal*. In the previous case of bounded losses, we provide a simple variant of the 1-Nearest-Neighbor (1NN) algorithm, which we call 2-Capped-1-Nearest-Neighbor (2C1NN), that is optimistically universal: it simply performs (1NN) on a restricted dataset by deleting points that have been used twice as nearest neighbor representatives. This rather small modification allows the algorithm to be universally consistent under all SMV processes and for general metric context and value spaces $\mathcal{X}$ and $\mathcal{Y}$.

1.2.2 Universal Regression with Adversarial Responses

In Chapter 4, we turn to the case where responses are noisy, arbitrarily correlated, or even adversarial, which encompasses to the standard supervised learning setting or non-parametric regression. Here, we also provide a complete characterization of learnable processes $\mathbb{X}$ and give optimistically universal algorithms. Unlike in the noiseless case, the class of learnable processes here depends on the value space $\mathcal{Y}$, unveiling a fundamental dichotomy. For most applications such as classification ($|\mathcal{Y}| < \infty$), regression ($\mathcal{Y} = \mathbb{R}$), or compact value spaces $\mathcal{Y}$, learnable processes for noiseless responses and arbitrarily correlated/adversarial responses coincide. Thus, in these cases, generalizing our results to noisy supervised learning comes at no additional cost: we combine the 2C1NN algorithm with sparsity and robustness techniques to overcome adversarial noise. For pathological value spaces $\mathcal{Y}$, however, such a generalization gap may exist and can be quantified. Intuitively, these correspond to value spaces for which even the mean-estimation problem of supervised learning without contexts ($\mathcal{X} = \{0\}$), cannot be solved to a fixed precision in a finite number of iterations. In that case, a more standard algorithm akin to empirical risk minimization is optimistically universal. These results significantly generalize seminal works on consistency for non-i.i.d. data that assumed ergodic stationarity or high-order moments of the responses [GO07].

1.2.3 Contextual Bandits and Optimistically Universal Learning

We investigate machine learning settings with partial feedback in Chapter 5. We focus on the contextual bandit setting, which is fundamental in sequential data-driven decision-making. This line of work could serve as a stepping stone for the study of universal learning in more general settings. In the contextual bandit problem, a decision-maker iteratively observes a context $x_t \in \mathcal{X}$, selects an action $a_t \in \mathcal{A}$, and then receives a reward r_t as a result of the context and action. For instance, a store may serve a sequence of customers, provide a list of product recommendations to each customer, and receive a reward if the recommendation leads to a purchase. The standard model for rewards in contextual bandits is that rewards, conditioned on the context and selected action, follow a fixed and time-invariant conditional probability distribution. The literature on universal learning in partial feedback settings is surprisingly sparse. In particular, prior to our work, it was unknown whether universal consistency in contextual bandits was possible even for finite actions and i.i.d. contexts $(X_t)_{t\geq 1}$, that is, whether we can provide an algorithm that would converge to the optimal policy in hindsight irrespective of how complex this target policy may be.

We provide optimistically universal algorithms in this stationary contextual bandit setting. The characterization of universally learnable processes here follows a trichotomy: whether the action space $\mathcal{A}$ is finite, countably infinite, or uncountable. Of particular interest, for finite action spaces $\mathcal{A}$, data processes SMV that were learnable with the full feedback of supervised learning can still be universally learned with the partial feedback of contextual bandits, which gives consistent algorithms under significantly weaker and provably-minimal assumptions on contexts compared to previous literature.

1.2.4 Adversarial Contextual Bandits

In Chapter 6, we challenge the classical stationarity assumption for contextual bandits. It is well understood indeed that environments may be non-stationary, or even adversarial in some applications. As a first important step towards more complex non-stationary machine learning settings such as reinforcement learning, we study in this chapter non-stationary rewards in contextual bandits.

Quite surprisingly, non-stationarity prohibits the existence of optimistically universal algorithms in contextual bandits for the main case of finite action spaces $\mathcal{A}$ and generic context spaces (including in particular all uncountable Polish spaces). This strongly contrasts with the full-feedback supervised learning setting from Chapters 3 and 4, where going from independent to adversarial rewards comes at no additional cost. Due to the need for exploration in contextual bandits, algorithms should balance between two very different strategies (1) *generalization*: rewards observed for all previous contexts provide information at the population level; and (2) *personalization*: specific contexts of interest may diverge significantly from the population and require individually tailored actions. On the conceptual level, these two objectives are incompatible in the face of non-stationarity without prior knowledge of the context process.

On the positive side, given sufficient prior knowledge on the distribution of contexts $(X_t)_{t\geq 1}$, we can still provide algorithms that find the right trade-off between generalization and personalization to achieve universal consistency. In fact, the class of universally learnable processes is still extremely large, beyond i.i.d. or ergodic processes, but it is in general strictly smaller than for contextual bandits with stationary rewards.

1.3 Memory Constraints in Optimization

We investigated in Part I the limits of learning with respect to generalizability and data assumptions, irrespective of the specific implementation of these algorithms. However, resource efficiency is also critical for practical implementations. In Part II, we specifically investigate the impact of memory in convex optimization.

The efficiency of optimization algorithms is most commonly analyzed through the lens of oracle complexity, which is the number of calls to an oracle (a black box that provides information about the function to optimize) needed for an algorithm to output an approximate solution within a desired tolerance error. With growing problem sizes, oracle complexity may not be the only bottleneck for optimization; in particular, practical constraints motivated the study of memory usage in algorithms and communication for decentralized optimization.

First-order convex optimization exemplifies memory usage challenges. Consider minimizing Lipschitz (non-smooth) convex functions on the d-dimensional unit ball to precision $\epsilon \leq 1/\sqrt{d}$, with access to a function value and gradient oracle. Two classes of algorithms are used to solve this problem in the literature. (1) Cutting-plane methods such as the center-of-mass method achieve the optimal oracle complexity $\mathcal{O}(d \ln \frac{1}{\epsilon})$. However, these are rarely used in high-dimensional applications and are often seen as impractical: cutting-plane methods require a large runtime per iteration. Moreover, these methods need to store all previously observed gradients, which requires at least a quadratic bit memory $\Theta(d^2 \ln \frac{1}{\epsilon})$.

(2) Gradient descent methods, on the other hand, are arguably the most convenient and commonly used methods. They only require storing and updating a few vectors, requiring $\mathcal{O}(d \ln \frac{1}{\epsilon})$ bit memory and runtime per iteration. As a major downside, however, their oracle complexity of $\mathcal{O}(1/\epsilon^2)$ is suboptimal.

This raises several natural questions. Can these algorithms be improved? More precisely, what is the trade-off between oracle complexity and memory usage? Here, we aim to provide some answers for convex optimization and related problems. For a concise summary of known results on the topic, we refer to Section 1.3.4.

1.3.1 Quadratic Memory is Necessary for Optimal Query Complexity in Convex Optimization

In Chapter 7, we start by showing some impossibility results in the oracle-complexity / memory landscape. We show that the full memory of cuttings planes is in fact necessary to achieve the optimal oracle complexity, at least for deterministic algorithms. Precisely, we show that for any $\delta \in [0, 1]$, any deterministic algorithm for convex optimization requires at least $d^{2-\delta}$ memory or make $\tilde{\Omega}(d^{1+\delta/3})$ oracle queries. The information-theoretic proof techniques build upon the work of [Mar+22] who first provided lower-bound trade-offs showing that having both optimal oracle complexity and optimal memory usage was impossible. We recall that because the optimal oracle complexity without memory constraints is $\mathcal{O}(d \ln \frac{1}{\epsilon})$, storing all gradient oracle information up to reasonable precision only requires $\mathcal{O}(d^2 \ln^2 \frac{1}{\epsilon})$[1] bits of memory, that is quadratic memory in the dimension d. Hence, our result implies that for deterministic algorithms, any non-trivial memory constraint ($\delta > 0$) strictly worsens the oracle complexity. In particular, cutting-plane methods such as the center-of-mass method are Pareto-optimal in the query-complexity/memory trade-off.

1.3.2 Memory-Constrained Algorithms for Convex Optimization

In the previous Chapter 7, we provided a lower-bound trade-off between oracle complexity and memory usage. These are impossibility results, which in particular leave open the major question of whether it is possible to improve over cutting planes or gradient descent. This is precisely the purpose of Chapter 8 in which we provide memory-efficient algorithms. Our proof techniques also generalize to the related *feasibility* problem in which one aims to find a point within a convex feasible set included in the unit d-dimensional ball, having only access to a separation oracle. This oracle either reports that a query point is already within the feasible set or provides a hyperplane that separates the query from the feasible set. For the feasibility problem with accuracy ϵ, we assume that the feasible set contains a ball of radius ϵ. One can easily check that this generalizes the convex optimization problem, where the gradient oracle can be used as a separation oracle for the feasible set of ϵ-minimizers.

The class of algorithms we introduce recursively use cutting-plane methods as subroutines on smaller-dimensional subproblems. For any parameter $p \in [d]$ which specifies the depth of the recursive construction, the algorithm requires $\mathcal{O}(\frac{d^2}{p} \ln \frac{1}{\epsilon})$ bits of memory and makes

[1] An extra $\ln \frac{1}{\epsilon}$ factor can be removed by restricting the search space using John's ellipsoid theorem, which gives exactly the memory $\mathcal{O}(d^2 \ln \frac{1}{\epsilon}$ required for cutting-plane methods.

$(C\frac{d}{p}\ln\frac{1}{\epsilon})^p$ oracle calls for some constant $C > 1$. In particular, $p = 1$ exactly corresponds to the standard cutting plane methods. In the sub-polynomial regime $\ln\frac{1}{\epsilon} \gg \ln d$, this class of algorithms provides a positive trade-off between cutting-planes and gradient-descent; to the best of our knowledge these are the first class of algorithms that are non-Pareto-dominated by the two classical optimization algorithms in any regime with $\epsilon \leq 1/\sqrt{d}$. Importantly, in the exponential regime $\epsilon \leq d^{-\Omega(d)}$, our algorithm with $p = d$ achieves the optimal memory usage and strictly improves the oracle-complexity of gradient descent from $\mathcal{O}(1/\epsilon^2)$ to $(C\ln\frac{1}{\epsilon})^d$.

1.3.3 Gradient Descent is Pareto-Optimal in the Oracle Complexity and Memory Trade-off for Feasibility Problems

The previous chapters left open two major questions. First, can we improve over gradient descent in standard accuracy regimes (outside the exponential regime $\epsilon \leq d^{-\Omega(d)}$)? Gradient descent is arguably the most commonly used method in practice, hence understanding the form of the memory trade-off near gradient descent is an important question to address. Second, while the lower bounds of Chapter 7 demonstrated the advantage of having larger memory, their separation in oracle complexity is very mild (at most an extra factor $\mathcal{O}(d)$ compared to the optimal oracle complexity). In particular, they do not show any dependency in the accuracy ϵ, which contrasts with the oracle complexity of gradient descent $\mathcal{O}(1/\epsilon^2)$. A natural question is therefore to understand what is the dependency in ϵ for the oracle-complexity of memory-constrained algorithms. In particular, given that cutting planes only exhibit a logarithmic dependency $\mathcal{O}(d\ln\frac{1}{\epsilon})$, can we escape a polynomial dependency in ϵ under memory constraints?

In Chapter 9 we give improved lower-bound trade-offs for the feasibility problem to answer these two questions. Precisely, we show that to solve feasibility problems with accuracy $\epsilon \geq e^{-d^{o(1)}}$, any deterministic algorithm either uses $d^{1+\delta}$ bits of memory or must make at least $1/(d^{0.01\delta}\epsilon^{2\frac{1-\delta}{1+1.01\delta}-o(1)})$ oracle queries, for any $\delta \in [0, 1]$. We prove similar (albeit weaker) results for randomized algorithms which imply that gradient descent is Pareto-optimal in the oracle complexity/memory trade-off. Further, the oracle complexity for deterministic algorithms is always polynomial in $1/\epsilon$ if the algorithm has less than quadratic memory in d: this reveals a phase transition since with quadratic $\mathcal{O}(d^2\ln\frac{1}{\epsilon})$ memory, the oracle complexity of cutting plane methods is only logarithmic in $1/\epsilon$.

1.3.4 Summary of Known Results in Oracle-Complexity/Memory Trade-offs in Convex Optimization

As a concise summary for this oracle-complexity/memory usage landscape in convex optimization and feasibility problems, we give illustrations of the currently known trade-offs.

High-dimensional regime $\epsilon \geq 1/\sqrt{d}$. In this regime, there is no trade-off between oracle-complexity and memory since gradient descent is already known to be optimal in oracle-complexity, even with the right constant factor [Nes03]. This regime corresponds to the case when the number of iterations is smaller than the dimension and will not be our focus here.

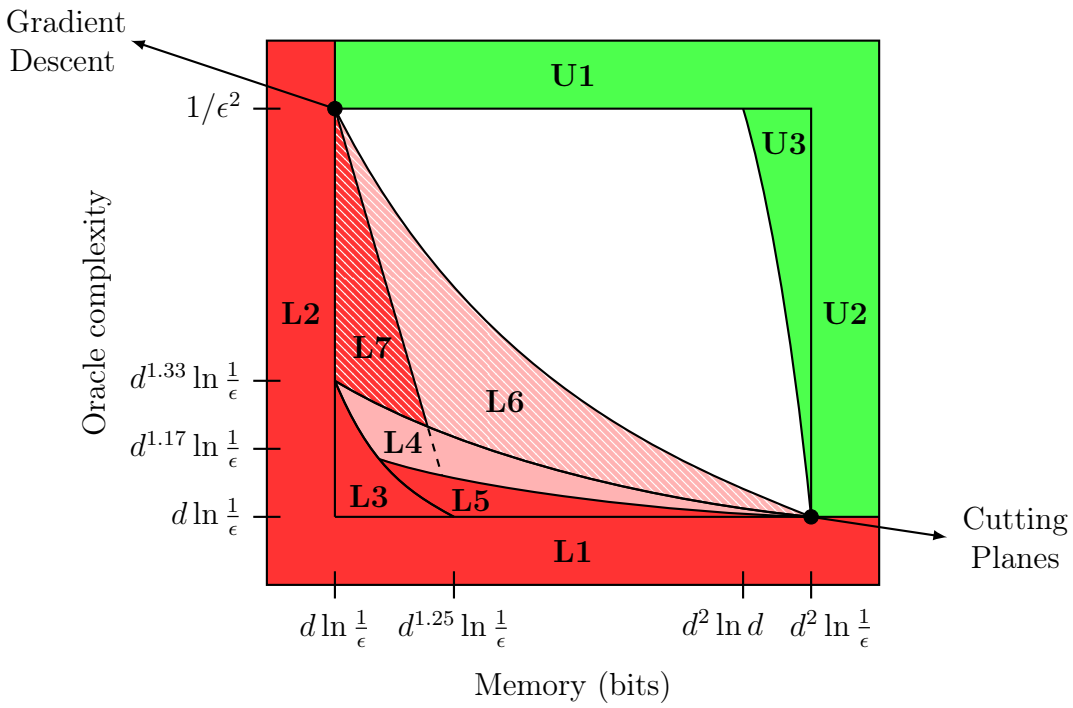

Figure 1.1: Oracle-complexity/memory trade-offs for convex optimization and the feasibility problem for the standard accuracy regime $1/\sqrt{d} \geq \epsilon \geq e^{-d^{o(1)}}$. The red (resp. pink) regions correspond to lower-bound regions that are not achievable by randomized (resp. deterministic) algorithms. Lower-bound regions that only hold for the feasibility problem are dashed. The green regions correspond to upper-bound regions that are achievable by (deterministic) algorithms for both the feasibility problem and convex optimization. References for each region are given in Section 1.3.4.

Standard accuracy regime $1/\sqrt{d} \geq \epsilon \geq e^{-d^{o(1)}}$. The most standard setting to study oracle-complexity/memory trade-offs is when the accuracy ϵ is not exponentially small in the dimension d; here we will consider that $\epsilon \geq e^{-d^{o(1)}}$. Fig. 1.1, summarizes both upper and lower bound trade-offs that were developed in [Mar+22; CP23] and in this thesis. Here the notation $\tilde{\Omega}$ hides poly-logarithmic factors in d. We start with the lower bounds.

L1 The optimal oracle complexity for convex optimization (a fortiori for the feasibility problem) is $\Theta(d \ln \frac{1}{\epsilon})$ [NY83].

L2 The optimal memory usage is $\Theta(d \ln \frac{1}{\epsilon})$ bits. This is in fact necessary even just to represent the output $x^\star$ of the algorithm with the desired accuracy ϵ, which can be proved with a simple covering argument [WS19].

L3 In a breakthrough, [Mar+22] showed the first lower-bound trade-off. Any algorithm for convex optimization uses $d^{1+\delta}$ bits of memory or has oracle-complexity $\tilde{\Omega}(d^{4/3(1-\delta)})$, for any $\delta \in [0, 1/4]$. Hence it is impossible to have both optimal memory usage and oracle-complexity.

L4 In Chapter 7 we show that any deterministic algorithm for convex optimization uses at least $d^{1+\delta}$ bits of memory or has oracle-complexity $\tilde{\Omega}(d^{4/3-\delta/3})$, for any $\delta \in [0,1]$. Hence, quadratic memory in the dimension d is necessary to have the optimal oracle-complexity for deterministic algorithms.

L5 [CP23] showed that any randomized algorithm for convex optimization uses at least $d^{1+\delta}$ bits of memory or has oracle-complexity $\tilde{\Omega}(d^{7/6-\delta/6-o(1)})$ whenever $\epsilon \leq e^{-\ln^5 d}$, for any $\delta \in [0,1]$. Hence, in this regime, quadratic memory is also necessary for randomized algorithms to have the optimal oracle-complexity.

- All previous lower-bounds can be adapted to include a factor $\ln \frac{1}{\epsilon}$ so that they always give non-trivial trade-offs, as we show in Chapter 8.

L6 Previous lower bounds held for convex optimization, hence a fortiori for the feasibility problem. For this harder problem, we give improved query lower bounds in Chapter 9. Any deterministic algorithm solving the feasibility problem uses $d^{1+\delta}$ bits of memory or has oracle-complexity $1/\left(d^{0.01\delta}\epsilon^{2\frac{1-\delta}{1+1.01\delta}-o(1)}\right)$, for any $\delta \in [0,1]$. In particular, the oracle-complexity of gradient descent is necessary to have the optimal memory usage.

L7 In Chapter 9, we also show that any randomized algorithm solving the feasibility problem uses $d^{1+\delta}$ bits of memory or has oracle-complexity $1/\left(d^{2\delta}\epsilon^{2(1-4\delta)-o(1)}\right)$, for any $\delta \in [0,1/4]$.

We then turn to the upper-bounds, that is, known algorithms for convex optimization or the feasibility problem.

U1 Gradient descent has optimal memory usage of $\Theta(d \ln \frac{1}{\epsilon})$ bits but has suboptimal $\mathcal{O}(1/\epsilon^2)$ oracle-complexity. It solves convex optimization, but also the feasibility problem which is perhaps less well-known: the vanilla algorithm with step-size ϵ converges with the same guarantees as for convex optimization.

U1 Cutting-plane methods use $\mathcal{O}(d^2 \ln \frac{1}{\epsilon})$ bits of memory but have the optimal oracle-complexity of $\mathcal{O}(d \ln \frac{1}{\epsilon})$. These are by nature designed to solve feasibility problems, hence a fortiori convex optimization problems as well.

U3 We introduce in Chapter 8 algorithms that provide some non-trivial trade-off between gradient descent and cutting-plane methods when $\ln \frac{1}{\epsilon} \gg \ln d$. The memory improvement depends on the accuracy. The memory required is divided by at most a factor $\ln \frac{1}{\epsilon} / \ln d$ and is more significant for smaller values of the tolerance accuracy ϵ.

Exponential accuracy regime. We next turn to the exponential regime $\epsilon \leq d^{-\Omega(d)}$, for which the upper-bound trade-off for the algorithms we introduce in Chapter 8 have different implications, which are represented in Fig. 1.2. The most significant differences are for the upper bounds. The upper bounds U1 and U2 are unchanged.

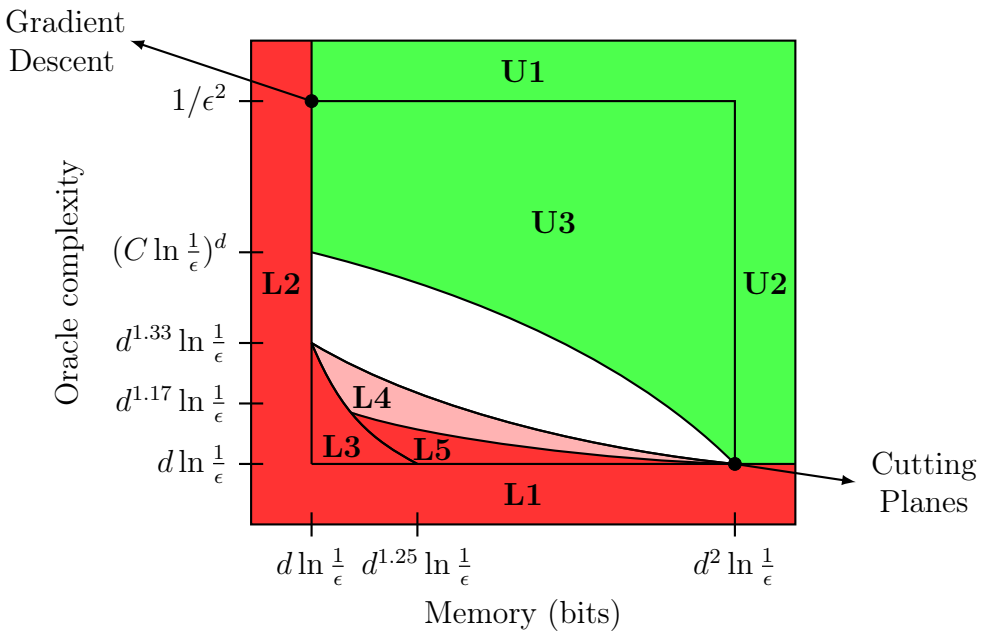

Figure 1.2: Oracle-complexity/memory trade-offs for convex optimization and the feasibility problem for the exponential accuracy regime $\epsilon \leq e^{-d^{\Omega(d)}}$. The red (resp. pink) regions correspond to lower-bound regions that are not achievable by randomized (resp. deterministic) algorithms. The green regions correspond to upper-bound regions that are achievable by (deterministic) algorithms for both the feasibility problem and convex optimization. References for each region are given in Section 1.3.4.

U3 In the exponential regime, the proposed algorithms from Chapter 8 strictly improve over gradient descent. This shows that gradient descent is not Pareto-optimal in that regime: we can improve its oracle-complexity from $1/\epsilon^2$ to $(C \ln \frac{1}{\epsilon})^d$ for some universal constant $C > 0$.

As for the lower-bound trade-offs, they all hold in the exponential regime except for L6 and L7. Indeed, these cannot hold since gradient descent is not Pareto-optimal anymore. We believe however that these could be adapted using the tools given in Chapter 8 to include dependencies in $\ln \frac{1}{\epsilon}$ for lower-bounds. In particular, it may be possible to give lower-bounds for the query-complexity of algorithms with optimal memory $\mathcal{O}(d \ln \frac{1}{\epsilon})$ of the form $(c_1 \ln \frac{1}{\epsilon})^{c_2 d}$ for some constants $c_1, c_2 > 0$. This would match the format of the upper-bound $(C \ln \frac{1}{\epsilon})^d$ which we provide in Chapter 8.

Strictly speaking, there is some gap between the standard regime $\epsilon \geq e^{-d^{o(1)}}$ and this exponential regime $\epsilon \leq d^{-\Omega(d)}$. In this case, the upper-bound U3 also holds but does not strictly improve over gradient descent. All lower-bounds L1 to L7 also hold but L6 and L7 do not reach gradient descent exactly. We refer to Chapter 9 for detailed bounds.

Part I

Universal Learning

Chapter 2

An Overview of Universal Learning

In this first chapter, we present a collection of results building a theory of universal learning for machine learning as first introduced by [Han21a]. In statistical learning theory, a major goal is to provide algorithms that can learn efficiently from observed data with provable performance guarantees for large classes of problem instances. Universal learning studies the fundamental question of *learnability* under provably minimal assumptions. Precisely, one aims to (1) precisely understand minimal assumptions on the problem instances necessary for learning, and (2) give algorithms with guarantees under these minimal assumptions.

2.1 A Gentle Introduction

The present section is meant to serve as a reader-friendly introduction to the universal learning framework. We give some context and motivation, and present the main questions.

2.1.1 Background on statistical learning.

To make this discussion more concrete, we present the universal learning approach for the sequential forecasting task which is a core building block for learning problems. As we will see, the framework naturally extends to more complex machine learning settings. In sequential forecasting, a learner iteratively makes predictions about a sequence of *values* $Y_1, Y_2, \ldots$ within a value space $\mathcal{Y}$, for instance predicting tomorrow's weather or future stock prices. To make its prediction, the learner typically has some side information X_t in some space $\mathcal{X}$, that hopefully is somewhat related to the value to be predicted Y_t. These are commonly referred to as *covariates* or *contexts*. In the weather forecasting example, covariates could include anything from pressure, temperature, humidity measurements, typical weather outcomes for that time of the year, among others. To summarize, at every time step $t \geq 1$, the learner has to make a prediction $\hat{Y}_t$ about the value Y_t using only the present context X_t and historical data, that is, past values $Y_1, \ldots, Y_{t-1}$ and past contexts $X_1, \ldots, X_{t-1}$.

To evaluate the performance of a learner, we use a *loss function* ℓ that measures the discrepancy $\ell(Y_t, \hat{Y}_t)$ between true and predicted values at any iteration. Then, ideally, one would aim to learn a (near-)optimal prediction function $\mathcal{X} \to \mathcal{Y}$ that given a context minimizes the expected loss. To give a useful example, in real-valued regression, a common

choice of loss function is the 2-norm $\ell(Y_t, \hat{Y}_t) = |Y_t - \hat{Y}_t|^2$. Further assuming that the context-value pairs $(X_t, Y_t)_{t\geq 1}$ are independent identically distributed (i.i.d.), under some integrability conditions, the optimal prediction function is the *Bayes predictor* $x \in \mathcal{X} \mapsto \mathbb{E}_{Y|X}[Y \mid X = x]$.

Two classes of problem restrictions. Considering the range of applications in which forecasting tasks arise, unsurprisingly, this problem has been extensively studied in the online learning and regression literature. To have any positive results on the problem, however, some assumptions about the problem are necessary. In particular, standard approaches make the following two classes of ad-hoc restrictions.

1. Instead of comparing the performance of the algorithm to the optimal predictions, one compares the algorithm to a pre-specified set of benchmark prediction functions $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$, also called a *hypothesis class*. As an example, in linear regression, the hypothesis class exactly corresponds to linear predictors f. Similarly, when training a machine learning model, the function class $\mathcal{F}$ that one aims to compete with is exactly prescribed by the form of the model, e.g., support vector machine separators, boosting tree functions, neural network functions, etc.

   This benchmark approach is sometimes referred to as the *agnostic* setting, in contrast with the *realizable* or *noiseless* setting in which one makes the significantly stronger assumption that the values Y_t are generated from the context X_t exactly through some function $f^\star$ within the function class $\mathcal{F}$, so that $Y_t = f^\star(X_t)$ for all $t \geq 1$.

2. Assumptions are made on the data generating process for the context sequence $\mathbb{X} = (X_t)_{t\geq 1}$ so that the past observed instances are sufficiently informative on future instances. The predominant assumption in the literature is that the contexts are i.i.d. but other assumptions such as ergodicity or stationarity can also be used. Similarly, assumptions can be made on how value process $\mathbb{Y} = (Y_t)_{t\geq 1}$ is generated with respect to the contexts $\mathbb{X}$. As described above, assuming that $(X_t, Y_t)_{t\geq 1}$ are together i.i.d. is common in regression settings.

To give an example of how such restrictions can be leveraged for learning, we give a brief overview of a central result in the Probably Approximately Correct (PAC) setting which was first introduced by [Val84]. This central framework in statistical learning will be useful to remember when introducing the universal learning framework.

PAC learning. Consider the binary classification setting where $\mathcal{Y} = \{0, 1\}$ and the loss is the $0-1$ loss, that is, our goal is simply to minimize the number of classification mistakes. As introduced above, we fix some hypothesis class $\mathcal{F} = \{f : \mathcal{X} \to \{0, 1\}\}$. We also assume that the context-value pairs (X_t, Y_t) are together i.i.d. according to some joint distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ unknown to the algorithm. The goal in PAC learning is, given T samples of $\mathcal{D}$, to output a prediction function $\hat{h} : \mathcal{X} \to \{0, 1\}$ such that with good probability $1 - \delta$ (probably), the proposed prediction function performs as well as the best function within the hypothesis class $\mathcal{F}$ (correct) up to an additional small fraction of mistakes ϵ (approximately).

As it turns out, whether this can be achieved without prior knowledge of $\mathcal{D}$ depends on the Vapnik-Chervonenkis (VC) dimension of the function class $\mathcal{F}$, which is a combinatorial complexity measure of the class. Its precise definition is not crucial for our present discussion but we give its definition here for completeness. We say that a finite set $S \subset \mathcal{X}$ is shattered by $\mathcal{F}$ if all $2^{|S|}$ functions $S \to \{0,1\}$ can be obtained by restricting some function from $f \in \mathcal{F}$ to the set S. The VC dimension of $\mathcal{F}$ is then the maximum cardinality of a set shattered by $\mathcal{F}$. For instance in linear regression, the function class of affine separators $\boldsymbol{x} \in \mathbb{R}^k \to \mathbb{1}[\boldsymbol{a}^\top \boldsymbol{x} \geq b]$ has VC dimension $k+1$. A central result in PAC learning is the following [VC71; SB14].

Theorem 2.1 (Fundamental Theorem of Statistical Learning, quantitive version)). *Fix a hypothesis class $\mathcal{F}$ with VC dimension $d < \infty$. There is some algorithm such that having access to T i.i.d. samples from any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, outputs a hypothesis $\hat{h}$ such that for any $\delta, \epsilon > 0$, with probability at least $1-\delta$,*

$$\mathbb{P}_{(X,Y)\sim\mathcal{D}}[\hat{h}(X) \neq Y] \leq \inf_{f\in\mathcal{F}} \mathbb{P}_{(X,Y)\sim\mathcal{D}}[f(X) \neq Y] + C\sqrt{\frac{d+\log(1/\delta)}{T}},$$

for some universal constant $C > 0$.

In particular, the excess loss compared to the best baseline within the hypothesis class decays to 0 as the number of samples T grows. The above result is tight up to constant factors in the excess loss term. Further, it is known that for a given benchmark hypothesis class $\mathcal{F}$, having finite VC dimension is necessary to have algorithms with vanishing excess loss. This fact that VC dimension characterizes PAC learnability is also known as the qualitative version of Theorem 2.1. Importantly, the algorithm that achieves the desired excess loss bound from Theorem 2.1 is arguably the simplest learning algorithm, Empirical Risk Minimization (ERM). It simply outputs the function $f \in \mathcal{F}$ that had the smallest in-sample loss, computed with the T available samples.

Of course, many generalizations of this result are possible in the loss function, the value spaces, or the considered setting. Our goal here is simply to give a brief overview of a useful framework that exemplifies both standard restrictions (1) on the class of functions $\mathcal{F}$ to compare the performance of the algorithm against, and (2) on the context sequence that is here supposed i.i.d. together with the values.

Trade-offs between assumptions in statistical learning. Ideally, one would like to use a benchmark function class $\mathcal{F}$ as large as possible so that we compare the performance of the algorithm to functions closer to the optimal predictions. However, one expects the guarantees to be weaker for larger benchmark function classes, since the learning objective is more demanding. This is exemplified in Theorem 2.1 which shows how the excess loss guarantee deteriorates as the VC dimension of the function class grows for the PAC learning setting. Similarly, other complexity measures for function classes have been extensively studied to quantify such trade-offs in other settings.

Alternatively, one could aim to understand the limits of *learnability*, that is what are minimal assumptions for learning guarantees to still be achievable. In particular, we aim to understand in which ways can we relax the two types of restrictions on benchmark function

classes and data-generating processes. Answers to this question reveal a fundamental trade-off between these two types of assumptions.

1. **No assumptions on $\mathbb{X}$.** On one extreme, consider the case when no assumptions are made on the context-generating process $\mathbb{X}$. This corresponds to the adversarial case in which an adversary can select contexts adaptively on the algorithm predictions [Lit88; LW94; CL06; BPS09; RST15a; Alo+21]. This setting is known to be quite restrictive. Even in the simpler realizable setting for binary classification $\mathcal{Y} = \{0, 1\}$, a classic result from [Lit88] showed that to have a decaying average number of mistakes, the considered function class $\mathcal{F}$ should have finite *Littlestone dimension*, in which case the number of mistakes is in fact bounded by this dimension $\mathrm{LDim}(\mathcal{F})$.[1] In the agnostic case, the number of mistakes can be bounded by $\mathcal{O}(\sqrt{\mathrm{LDim}(\mathcal{F})T})$ [BPS09; Alo+21]. This combinatorial complexity measure is much more restrictive than the VC dimension, for instance even 1-dimension threshold functions have infinite Littlestone dimension. A fortiori, linear regression with adversarial contexts is impossible.

2. **Assumptions on both $\mathbb{X}$ and $\mathcal{F}$.** To learn beyond finite Littlestone dimension function classes, many assumptions on $\mathbb{X}$ have therefore been proposed. As mentioned above, most works assume that the sequence of context-value pairs $(X_t, Y_t)_{t\geq 1}$ is i.i.d.This is the classical setting for parametric and non-parametric regression [Gyö+02; Was06] for which extensive results are known for various types of function class estimators, including but not limited to kennel smoothing, local polynomials [Tsy09], smoothing splines [DD78; GS93; Wah90], reproducing kernel Hilbert spaces [SS02; Wah90] or wavelets [Mal99]. In the PAC learning setting, learning is possible exactly for function classes with finite VC dimension d, with tight excess loss bounds of $\mathcal{O}(\sqrt{(d + \log(1/\delta)T})$ for the agnostic case [VC71] as shown in Theorem 2.1, and $\mathcal{O}(d + \log(1/\delta))$ for the realizable case [Vap82; Blu+89; HLW94; Han16]. Beyond PAC learning, many other joint assumptions on the data generating process $\mathbb{X}$ and classes of functions relating contexts to values have been considered, including [Rya06; UB13; Bou+21].

3. **No function class restrictions: universal learning.** Sitting at the other extreme of the trade-off, this third category is what we loosely refer to as *universal learning*. This corresponds to a setting in which the dependency between the covariates and the corresponding values can be arbitrary. We give separately a more in-depth overview of the universal learning literature.

2.1.2 Previous results in universal learning.

In universal learning, all assumptions on the prediction function to be learned are lifted. Hence, the goal becomes to understand which assumptions on the context-generating process can be used to preserve learning guarantees.

[1] The Littlestone dimension is defined as the maximal depth of a full binary tree with inner nodes labeled by contexts in $\mathcal{X}$ and such that for every path from the root to a leaf there is a function in class $f \in \mathcal{F}$ that evaluated on each inner node context evaluates to 0 (resp. 1) if the path follows the left (resp. right) child.

Universal learning with i.i.d. contexts. Initial works on universal learning considered the case of i.i.d. context sequences $\mathbb{X}$. The first early in this direction showed that for binary classification $\mathcal{Y} = \{0, 1\}$ in Euclidean context spaces and for the realizable case, we can achieve a sublinear number of errors *for any* target value function $f^\star : \mathcal{X} \to \{0, 1\}$ [CH67; Sto77; DGL13]. Algorithms that achieve this property are said to be *universally consistent*: consistent because they eventually learn the true prediction function, and universal because they achieve the desired property irrespective of how complex the true prediction function may be. As it turns out, in this realizable setting, in Euclidean context space, the simple 1-nearest neighbor algorithm is already universally consistent [CH67; Sto77; DGL13].

We spell out this seminal result for the sake of clarity. We still consider the sequential binary classification prediction problem but further assume that the sequence $\mathbb{X}$ is i.i.d. and that the values are given via $Y_t = f^\star(X_t)$ at every time step, which corresponds to noiseless responses. Importantly, no assumptions are made on $f^\star$ other than it is measurable. At every iteration, the k-nearest neighbor works as follows: given a current context X_t and past historical data $(X_{t'}, Y_{t'})_{t'<t}$, it first computes the k nearest neighbors of X_t within the past contexts (with the ambient metric of $\mathcal{X}$), say $X_{t_1}, \dots, X_{t_k}$, then outputs the majority vote over their values $Y_{t_1}, \dots, Y_{t_k}$. Then, for any i.i.d. sequence $\mathbb{X}$ on $\mathcal{X} = \mathbb{R}^d$ and irrespective of the target $f^\star : \mathcal{X} \to \{0, 1\}$, the predictions $\hat{Y}_t$ of the 1-nearest neighbor algorithm satisfy

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[\hat{Y}_t \neq f^\star(X_t)] = 0, \quad (a.s).$$

Beyond the realizable case, for i.i.d. context-value pairs $(X_t, Y_t)_{t\geq 1}$ according to some distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, [Dev+94] showed that the k_t-nearest neighbor rule with $k_t/\log t \to \infty$ and $k_t/t \to 0$ is also consistent in the agnostic setting under mild integrability assumptions. In this noisy setting, one cannot possibly reach a zero average error rate. The minimum error sometimes referred to as the *Bayes risk* is defined as

$$R_{\mathcal{D}} := \inf_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(X,Y)\sim\mathcal{D}} \left[\ell(f(X_t), Y_t)\right],$$

where the infimum is taken over all measurable functions. In the binary classification setting, the optimal risk is simply $R_{\mathcal{D}} = \mathbb{E}_X \min\left\{\mathbb{P}(Y = 1 \mid X), \mathbb{P}(Y = 0 \mid X)\right\}$. The goal here is then to reach the minimal Bayes risk, that is, ensure that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) = \mathcal{R}_{\mathcal{D}}, \quad (a.s).$$

Both previous results were tailored to Euclidean context spaces. More recently, [Han+21; GW21] showed that these consistency results can be extended to general *separable* metric spaces, that is, admit a countable and dense set. In fact, [Han+21] shows that universal consistency under i.i.d. contexts is possible exactly for context metric spaces $\mathcal{X}$ that are *essentially-separable*[2], a notion that slightly generalizes separable spaces. For our discussion, working with separable context spaces will be largely enough. Following these results, [TK22] showed how to generalize these results beyond binary classification for general separable value spaces $\mathcal{Y}$ and under sufficient integrability conditions on the coupling distribution $\mathcal{D}$.

[2]A metric space $(\mathcal{X}, \rho)$ is essentially-separable if for every probability measure μ on the Borel measure induced by ρ, there is a subspace $\mathcal{X}' \subseteq \mathcal{X}$ with full measure $\mu(\mathcal{X}') = 1$ and such that $(\mathcal{X}', \rho)$ is separable.

Qualitative comparison to PAC learning. The previous discussion essentially shows that under i.i.d. contexts, universal learning can always be achieved. A few remarks are in order before moving to non-i.i.d. contexts. In the PAC learning setting, we saw that learning was only possible if the benchmark function class had finite VC dimension. On the other hand, in the universal learning framework, we seek guarantees compared to the optimal prediction functions, say the Bayes predictor if it exists. We stress that the nature of the results in the universal learning setting is, however, very different. While PAC learning gives error bounds for a fixed sample horizon and for the *worst-case* learning distribution $\mathcal{D}$; in universal learning, one first *fixes* the learning distribution $\mathcal{D}$, then as the number of samples T grows for that distribution, one aims to have asymptotic convergence. In particular, this takes advantage of the fact that the worst-case learning distribution for t_1 samples may not be the worst-case distribution for $t_2 > t_1$ samples. The previous universality results precisely show that for every distribution, given enough samples one can reach the optimal Bayes risk. Since giving uniform convergence rates is impossible without imposing restrictions on the target prediction functions considered, the universality results that we present here are therefore very *asymptotic* in nature.

Additionally, we note that the idea of first fixing the learning problem and then making the number of samples grow proved useful to obtain much faster rates of convergence [Bou+21] than in PAC learning where the worst-case learning problem depends on the number of samples.

Universal learning beyond i.i.d. contexts. Going beyond i.i.d. processes, the notion of universal consistency needs to be slightly adapted, since a minimal risk as in the i.i.d. case may not necessarily exist. We recall that the main goal of universal learning is to unrestrict the class of benchmark functions. That is, we want to ensure that eventually the predictions of the algorithm $\hat{Y}_t$ perform at least as well as those of any prediction function $f : \mathcal{X} \to \mathcal{Y}$. Hence, in its strongest form, our objective is to ensure

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 0, \quad (a.s.), \quad \forall f : \mathcal{X} \to \mathcal{Y}.$$

Non-i.i.d. context sequences have also been extensively studied in the literature. These works use relaxations of the i.i.d. assumption, including mixing conditions [SHS09; LKS06; Rou88; Col84; Irl97], stationary ergodicity [Orn78; Alg92; MYG96; GL02; Nob03], or certain forms of law of large numbers [MKN99; GG09; SHS09], to achieve similar universality results. We note however that most of these assumptions are relaxations that in essence still allow to use technical tools from the i.i.d. case to generalize the universality results. In particular, these assumptions on the context-generating process are still *ad-hoc*: there are possibly many other choices of classes of processes for which universal learning can be performed. To give an example, one could imagine that stochastic processes that alternate between different (in finite number) distributions should still be learnable, although these would neither be stationary nor ergodic. Understanding which are the fundamental and exact assumptions required for learning is precisely the goal of the so-called *optimistic learning* framework.

2.1.3 Optimistic learning: Minimal assumptions for learning

[Han21a] was the first to introduce a systematic framework to understand learning with minimal assumptions. Instead of using *ad-hoc* relaxations of the i.i.d. assumption that still allow to preserve technical parts of the analysis for the i.i.d. case, [Han21a] proposed the much more ambitious goal of characterizing the *provably-minimal* assumptions for which universal learning is achievable, that is, understanding when universal learning is possible. This corresponds to characterizing the following class of universally learnable stochastic processes

$$\mathcal{C} := \{\text{processes } \mathbb{X} : \exists \text{ an universally consistent algorithm provided the contexts follow } \mathbb{X}\}.$$

To give an example, consider the simplest realizable setting, in which we assume that the values are always given as $Y_t = f^\star(X_t)$ for some unknown true prediction function $f^\star$. Then, we can explicit the definition of universally learnable processes above as follows.

$$\mathcal{C} = \left\{ \text{processes } \mathbb{X} : \exists \text{ an algorithm such that } \forall f^\star : \mathcal{X} \to \mathcal{Y}, \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) = 0 \right\}.$$

In particular, we saw previously that i.i.d. and stationary ergodic processes belong to $\mathcal{C}$. On the other hand, in general, not all stochastic processes are universally learnable. We give here a useful example to understand the difficulties in universal learning. Consider the simplest realizable binary classification setting $\mathcal{Y} = \{0, 1\}$ and suppose that the context space is simply $\mathcal{X} = \mathbb{N}$. The deterministic process $(X_t = t)_{t\geq 1}$ is *not* universally learnable. The reason is rather simple. At the t-th iteration, an algorithm only has access to the value of the function $f^\star$ on $\{1, \dots, t-1\}$. But because we impose no conditions on the true function $f^\star$, these do not give any information on the value $f^\star(t) \in \{0, 1\}$. Hence, no prediction is better than a random guess and the algorithm incurs a large average loss. Generalizing this example, we see that for any infinite context space $\mathcal{X}$, deterministic processes that regularly visit new contexts will not be universally learnable either.

By definition, there is no hope of having universally consistent algorithms if the context process $\mathbb{X}$ falls outside of the universally learnable processes $\mathcal{C}$. Assuming that $\mathbb{X} \in \mathcal{C}$ is therefore the *provably-minimal* assumption that one could hope to have positive results for. This leads us to our next question of interest that was succinctly stated in [Han21a] as: *Can we learn whenever learning is possible?* Precisely, we would like to provide algorithms that universally learn under this minimal assumption $\mathbb{X} \in \mathcal{C}$ that universal learning is achievable. For reasons that we will explain shortly, these are called *optimistically universal* algorithms. If these exist, which is not obvious a priori, then these algorithms are the most general possible in universal learning. In particular, they enjoy the strong property that for any process $\mathbb{X}$, if they are not universally consistent, then no other algorithm would be either.

To make the definition of optimistically universal algorithms clear, we emphasize that compared to the definition of universally learnable processes $\mathcal{C}$ there is a key interversion in quantifiers. For a process $\mathbb{X}$ to be universally learnable it suffices that there is some algorithm, tailored for $\mathbb{X}$, that is universally consistent under $\mathbb{X}$. On the other hand, an optimistically universal algorithm must be universally consistent under every such process.

The optimist's decision theory [Han21a]. The framework that was just introduced is part of a much more general—almost philosophical—form of reasoning that was introduced and named by [Han21a] as the optimist's decision theory. Imagine that we have fixed some sort of objective O, which can be a learning guarantee such as consistency or specific convergence rates, but potentially much more general. The minimal assumption to apprehend the task is that it is at least possible to solve it with some (unknown) methodology. Of course, not all tasks may be solvable, just as not all stochastic processes are universally learnable. Further, checking whether a specific task is solvable likely cannot be practically checked; deciding to apprehend the task is mostly a leap of faith. Hence this minimal assumption is aptly called the *optimist's assumption*. We are then interested in finding methodologies that solve all tasks for which the objective O can be possibly achieved. These are precisely *optimistic* methodologies that only rely on the optimist's assumption.

In our specific universal learning setting, the objective O is universal consistency, and optimistic methodologies are exactly optimistically universal algorithms.

2.1.4 Other universal learning settings

We presented the universal learning framework for online sequential prediction, but this can be generalized to many other machine learning models. In particular, we will also investigate contextual bandit settings for which the type of feedback is weaker than in regression settings. Beyond online learning, [Han21a] also characterized universal learnability for inductive and self-adaptive learning, two variants of the online learning setting. In online learning, the algorithm can indefinitely update its predictions based on observed data. Instead, in the inductive learning setting, the algorithm observes T samples $(X_t, Y_t)_{t\leq T}$ and then has to commit to a prediction function for all future iterations. We evaluate the performance of the algorithm as the number of available samples T grows to infinity. The self-adaptive setting lies in between the inductive and online learning: the algorithm also observes T samples $(X_t, Y_t)_{t\leq T}$ and can also adapt to the following sequence of covariates $X_{T+1}, X_{T+2}, \cdots$, but not to the values $Y_{T+1}, Y_{T+2}, \cdots$. We refer to [Han21a] for a detailed presentation of the known results for universal learning in these alternative learning settings.

2.2 Summary of Known Results in Universal Learning

In this section, we present the currently known results on universal online learning. These span three classical machine learning settings: realizable or noiseless settings, regression with arbitrary noise settings, and contextual bandit settings. Quite surprisingly, for most of these settings, we can give precise answers to the two main questions described above: (1) What is the class of universally learnable processes? (2) Can we construct optimistically universal algorithms, if they exist? As a brief preview, universal consistency can indeed be achieved for very general classes of processes, well beyond i.i.d. or stationary ergodic processes. Further, with the notable exception of adversarial contextual bandits, we can always give optimistically universal learning algorithms. In other terms, we can indeed learn whenever learning is possible.

2.2.1 Formal setup

We now formally present the universal learning setup. We mostly take the viewpoint of regression settings which were also used as motivation in the previous section, however, most of this setup will be shared for the contextual bandit setting as well.

Context and value space. The context space $(\mathcal{X}, \rho)$ is a general separable metric space taken with its Borel topology $\mathcal{B}$ induced by ρ. We recall that separability means that there exists a countable set that is dense with respect to the ambient metric. Similarly, the value space $(\mathcal{Y}, |\cdot|)$ can be any separable metric space. The loss $\ell : \mathcal{Y}^2 \to [0, \infty)$ can be more general than the ambient metric $|\cdot|$ but some assumptions are still required.

Definition 2.1 (Near-metrics and generalized-metrics)**.** *We say that a function $\ell : \mathcal{Y}^2 \to [0, \infty)$ is a* near-metric *if it is (1) symmetric: $\ell(y_1, y_2) = \ell(y_2, y_1)$ for all $y_1, y_2 \in \mathcal{Y}$; (2) positive: $\ell(y_1, y_2) = 0$ if and only if $y_1 = y_2$; and (3) satisfies the following relaxed triangular inequality,*

$$\forall y_1, y_2, y_3 \in \mathcal{Y}, \quad \ell(y_1, y_3) \leq c_\ell(\ell(y_2, y_1) + \ell(y_2, y_3)).$$

We say that a function $\ell : \mathcal{Y}^2 \to [0, \infty)$ is a generalized-metric *if it is (1) symmetric, (2) positive, and (3) satisfies the following slightly stronger relaxed triangular inequality,*

$$\forall \epsilon > 0, \exists C_\epsilon \geq 0, \forall y_1, y_2, y_3 \in \mathcal{Y}, \quad \ell(y_1, y_3) \leq (1 + \epsilon)\ell(y_2, y_1) + C_\epsilon \ell(y_2, y_3).$$

Unless mentioned otherwise, we suppose that ℓ is a near-metric. For regression settings, this may not be sufficient and we will then assume that the loss is a generalized-metric. The main point is that both are general enough to include p-losses: $\ell = |\cdot|^p$ for any $p > 0$, that are ubiquitous in machine learning. We denote by $\bar{\ell} := \sup_{y_1, y_2 \in \mathcal{Y}} \ell(y_1, y_2)$ the loss function supremum and say that the loss is bounded if $\bar{\ell} < \infty$. As we will see this will be the main case of interest for universal learning.

Online learning setup. We consider the following sequential learning problem. At every step $t \geq 1$, the learner observes a new instance $X_t \in \mathcal{X}$, predicts a value $\hat{Y}_t \in \mathcal{Y}$ then receives some form of observation $O_t \in \mathcal{O}$. In the realizable and regression settings, the learner observes exactly the true value $O_t = Y_t$ and the observation space is simply the value space $\mathcal{O} = \mathcal{Y}$. We keep this level of generality in the observations because in contextual bandits, the observation is of a different nature: we only get to observe a reward $O_t = r_t$ typically in $\mathcal{O} = [0, \infty)$. More importantly, the algorithm can *only* use the observed history to make its predictions. So far we use the term algorithm for simplicity. We note however that we allow the predictions to be as complex as desired: computability will not be a concern here although the algorithms we will propose will be somewhat implementable. Hence, we use the preferred term *learning rule* from now on.

Definition 2.2 (Learning rule)**.** *A* learning rule *is a sequence $f_\cdot = (f_t)_{t \geq 1}$ of possibly randomized measurable functions $f_t : \mathcal{X}^{t-1} \times \mathcal{O}^{t-1} \times \mathcal{X} \to \mathcal{Y}$. The value selected at time t by the learning rule is $\hat{Y}_t = f_t((X_s)_{s \leq t-1}, (O_s)_{s \leq t-1}, X_t)$.*

Consistency with general data generating processes. Since we are interested in general data-generating processes, we model the sequence of contexts $\mathbb{X} := (X_t)_{t\geq 1}$ and values $\mathbb{Y} := (Y_t)_{t\geq 1}$ as general stochastic processes on $(\mathcal{X}, \rho)$ and $(\mathcal{Y}, \ell)$ respectively. Depending on the learning setup, the model for how the responses $\mathbb{Y}$ and observations $\mathbb{O} = (O_t)_{t\geq 1}$ are related to the contexts $\mathbb{X}$ can be quite different. This will be formally defined within the corresponding Sections 2.2.2 to 2.2.5 below. To give an example, in realizable online learning, we assume that $O_t = Y_t = f^\star(X_t)$ for all $t \geq 1$ where $f^\star : \mathcal{X} \to \mathcal{Y}$ is some arbitrary measurable function. For convenience, we will use the notation $\mathbb{X}_{\leq t} = (X_{t'})_{t'\leq t}$ to denote the first t elements of the stochastic process.

We last formally define our learning objective. We specifically focus on *consistent* learning rules that achieve low long-run average loss compared to any fixed prediction function. Of course, many other objectives are possible, but consistency is perhaps the most fundamental property that reasonable algorithms should possess for learning. Precisely, we ask that the predictions of the algorithm always have vanishing excess loss compared to any measurable prediction function.

Definition 2.3 (Consistency)**.** *Let $(\mathbb{X}, \mathbb{Y}, \mathbb{O})$ be a stochastic process on $\mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, where $\mathbb{O} = (O_t)_{t\geq 1}$ are the learner's feedback observations. Let f. be a learning rule and denote by $\hat{Y}_t = f_t(\mathbb{X}_{\leq t-1}, \mathbb{O}_{\leq t-1}, X_t)$ its prediction at time $t \geq 1$.*

We say that f. is consistent under $(\mathbb{X}, \mathbb{Y})$ if for any measurable function $g : \mathcal{X} \to \mathcal{Y}$,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(g(X_t), Y_t) \leq 0, \quad (a.s.).$$

This notion generalizes the standard notions of consistency that we mentioned in the introduction. In the simplest realizable setting, this exactly asks for the average loss of the learner to decay to 0; in the classical regression setting when $(X_t, Y_t)_{t\geq 1}$ is i.i.d., this asks for the average loss to converge to the minimum Bayes risk.

2.2.2 Realizable (noiseless) learning

This is perhaps the simplest and classical setting in online learning. The main realizability assumption is that there is some underlying true function $f^\star : \mathcal{X} \to \mathcal{Y}$ that perfectly fits the data. Hence, at every iteration $t \geq 1$, the learner exactly observes the true value which is computed as $O_t = Y_t = f^\star(X_t)$. In particular, the responses are noiseless.

Our goal in universal learning is to provide learning rules that are consistent whatever the true underlying function $f^\star$, irrespective of how complex it may be. This leads us to the following definition of universal consistency.

Definition 2.4 (Universal consistency (realizable case))**.** *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$ and f. a learning rule. We say that f. is* universally consistent *under $\mathbb{X}$ if for any measurable function $f^\star : \mathcal{X} \to \mathcal{Y}$, the learning rule f. is consistent for the contexts $\mathbb{X}$ and values $Y_t = f^\star(X_t)$ for $t \geq 1$, equivalently,*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, f^\star(X_t)) = 0, \quad (a.s.),$$

where $\hat{Y}_t = f_t(\mathbb{X}_{\leq t-1}, (f^\star(X_{t'}))_{t'\leq t-1}, X_t)$ *is the prediction made at time* t*.*

We next define the set SOUL[3] (Strong Online Universal Learning) of universally learnable processes for which universal consistency can be achieved by some learning rule.

$$\text{SOUL} := \{\text{processes } \mathbb{X} : \exists \text{ learning rule } f. \text{ universally consistent under } \mathbb{X}\}.$$

Remark 2.1. *In the naming of* SOUL*, the term "strong" refers to the fact that we seek consistency guarantees in the* almost sure *sense. It is possible to study the universal learning questions for the weaker objective of consistency* in expectation*. The intuitions of the results are mostly the same however, hence we will only present results for strong consistency here.*

We are interested in learning rules that achieve universal consistency whenever possible.

Definition 2.5 (Optimistically universal learning rules)**.** *A learning rule* $f.$ *is* optimistically universal *if it is universally consistent under all processes* $\mathbb{X} \in$ SOUL*.*

Bounded losses. We are now ready to present the results for realizable online learning. We first introduce the class of process that will characterize SOUL. We refer to this condition as SMV (Sub-linear Measurable Visits). Intuitively, it asks that for any measurable partition of the input space $\mathcal{X}$, the process $\mathbb{X}$ only visits a sublinear number of its regions.

Condition SMV. *For every disjoint sequence* $\{A_k\}_{k=1}^\infty$ *of measurable sets of* $\mathcal{X}$ *such that* $\bigcup_{k=1}^\infty A_k = \mathcal{X}$*, (every countable measurable partition),*

$$|\{k \geq 1 : A_k \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T), \quad (a.s.).$$

By abuse of notation, let SMV be the collection of processes $\mathbb{X}$ *satisfying this condition.*

Here the intersection $A_k \cap \mathbb{X}_{\leq T} \neq \emptyset$ simply checks whether A_k has been previously visited during the first T iterations: $\mathbb{X}_{\leq T}$ should rather be interpreted as $\{X_1, \ldots, X_T\}$. While this may not be completely obvious, this is a very general class of processes, that includes i.i.d., stationary, ergodic processes but also significantly generalizes these.

Our main result for the realizable case is that SMV characterizes SOUL at least for bounded losses and we can also give optimistically universal learning rules.

Theorem 2.2 (Bounded losses [Bla22])**.** *Fix a separable metric space* $(\mathcal{X}, \rho)$*, and a separable near-metric space* $(\mathcal{Y}, \ell)$ *with bounded loss* $\bar{\ell} := \sup_{y_1,y_2} \ell(y_1, y_2) < \infty$*. Then* SOUL $=$ *SMV and a simple algorithm called 2C1NN (2-Capped-1-Nearest-Neighbor) is optimistically universal.*

The algorithm 2C1NN is a simple variant of the classical 1-Nearest-Neighbor (1NN) algorithm, which performs 1NN over a restricted dataset. More precisely, it is designed to ensure that the number of times each datapoint is used as a nearest neighbor is capped by 2: once a datapoint X_t has been used as a nearest neighbor twice, it is deleted from the training dataset. For context, although in Euclidean context spaces $\mathcal{X} = \mathbb{R}^d$ the 1NN algorithm is

[3]Throughout this document, we use the small-caps font style to denote universally learnable classes.

universally consistent [CH67; Sto77; DGL13] for i.i.d. processes $\mathbb{X}$, it is not optimistically universal. In fact, this would be the case for all other kNN algorithms even for $\mathcal{X} = [0, 1]$.

Further, kNN methods were known to fail in non-Euclidean spaces: a classical construction that can be found in [CG06] showed that in some infinite-dimensional subspaces there exists a distribution μ and a measurable set A such that $\mu(A) = 1/2$, but locally, the set A is "invisible" the kNN algorithms:

$$\lim_{\epsilon \to 0} \frac{\mu\{A \cap B(x;\epsilon)\}}{\mu\{B(x;\epsilon)\}} = 0, \quad \text{for } \mu\text{-almost every } x \in \mathcal{X}.$$

Here $B(x;\epsilon) = \{x' \in \mathcal{X} : \rho(x, x') \leq \epsilon\}$ is the closed ball centered at x of radius ϵ. At a very high level, the 2C1NN algorithm circumvents these problematic cases by deleting problematic data points regularly.

Unbounded losses. As it turns out, universal learning for unbounded losses is very restrictive. Because the losses can be arbitrarily large, making even a single mistake can be fatal if our goal is to have a vanishing average loss. The class of processes that arises in this case are FS (Finite Support) processes that only visit a finite number of distinct contexts.

Condition FS. *The process $\mathbb{X}$ satisfies $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X} \neq \emptyset\}| < \infty$ (a.s.). By abuse of notation let FS be the collection of processes satisfying this condition.*

Note that in this noiseless setting, learning under $\mathbb{X} \in$ FS processes is straightforward: it suffices to memorize the values observed for each distinct context. This memorization algorithm only incurs a finite number of mistakes because of the FS property, hence will be consistent. The main result is that FS characterizes Soul, that is, we cannot universally learn beyond these processes.

Theorem 2.3 (Unbounded losses [Han21a; BCH22])**.** *Fix a separable metric space $(\mathcal{X}, \rho)$, and a separable near-metric space $(\mathcal{Y}, \ell)$ with unbounded loss $\bar{\ell} := \sup_{y_1, y_2} \ell(y_1, y_2) = \infty$. Then Soul = FS and memorization is optimistically universal.*

This is a rather negative result since FS processes are extremely restrictive. Most i.i.d. processes (for non-atomic distributions) never visit the same context twice hence universal learning with unbounded losses would be impossible for these. Alleviating this negative result with added integrability assumptions is however possible as we will later see.

2.2.3 Regression: standard supervised learning

We now consider general regression settings in which the observation at time t is still the true value Y_t, but without making the realizable assumption. Contrary to the noiseless setting in which we assumed that ℓ was a near-metric, we need here to assume that it is a generalized-metric, which is slightly stronger (see Definition 2.1). The framework is general enough to incorporate arbitrary correlations between the context sequence $\mathbb{X}$ and the responses $\mathbb{Y}$: we allow the pair $(\mathbb{X}, \mathbb{Y})$ to be a general stochastic process on the product space $\mathcal{X} \times \mathcal{Y}$. These correspond to the arbitrarily-dependent responses defined by [Han22]. We can also allow the responses to be *adversarial*, that is the value Y_t can also depend on the past

predictions $\hat{Y}_{t'}$ for $t' < t$ and any internal randomness used to construct these. We refer to Chapter 4 for the exact definitions dealing with measure-theoretic concerns. In either case, the universal learning characterizations turn out to be the same for both adversarial and arbitrarily-dependent responses. We will refer to the corresponding processes $(\mathbb{X}, \mathbb{Y})$ as an adversarial process. In this context, the same definition of universal consistency as in Definition 2.4 becomes the following, as per the definition of consistency from Definition 2.3.

Definition 2.6 (Universal consistency (general regression))**.** *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$ and f. a learning rule. We say that f. is* universally consistent *under $\mathbb{X}$ for adversarial responses if it is consistent under any adversarial process $(\tilde{\mathbb{X}}, \mathbb{Y})$ with $\tilde{\mathbb{X}} \sim \mathbb{X}$.*

Similarly as in the realizable case, the set of universally learnable processes SOLAR (Strong universal Online Learning with Adversarial Responses) is defined as

$$\text{SOLAR} = \{\text{processes } \mathbb{X} : \exists \text{ learning rule } f. \text{ universally consistent under } \mathbb{X}\}$$

and we have the same definition for optimistically universal learning rules as in Definition 2.5 by simply replacing SOUL with SOLAR.

In this setting, we can again characterize exactly the class of learnable processes SOLAR. Conveniently, in most cases this turns out to still be SMV, hence learning under adversarial responses came at no expense compared to learning with noiseless responses for which SOUL = SMV. However, in some pathological regression settings, we may not be able to universally learn under all SMV processes. In these cases, a new condition CS (Continuous Submeasure) arises. For a context process $\mathbb{X}$ and any measurable set $A \in \mathcal{B}$ of $\mathcal{X}$, let $\hat{\mu}_{\mathbb{X}}(A) := \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[X_t \in A]$ be the long-run proportion of times the process visits A. The condition asks that $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(\cdot)]$ is a continuous sub-measure as follows.

Condition CS. *For every decreasing sequence $\{A_k\}_{k=1}^{\infty}$ of measurable sets in $\mathcal{X}$ with $A_k \downarrow \emptyset$,*

$$\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_k)] \underset{k\to\infty}{\longrightarrow} 0.$$

By abuse of notation, let CS denote the collection of processes $\mathbb{X}$ satisfying this condition.

This condition, while more restrictive than SMV, is still very general and includes all i.i.d., stationary, or ergodic processes. As a remark, these turned out to be the characterization of universally learnable processes in noiseless inductive and self-adaptive learning [Han21a] which we briefly mentioned in Section 2.1.4.

The alternative whether SOLAR = SMV or SOLAR = CS only depends on the value space $(\mathcal{Y}, \ell)$, but not on the context space $\mathcal{X}$. We call the property that characterizes this alternative F-TiME (Finite-Time Mean Estimation). It essentially asks that estimating the mean (or rather the Fréchet mean) of a sequence can be done in an online fashion to any arbitrary precision in finite time.

Property F-TiME. *For any $\eta > 0$, there exists a horizon time $T_\eta \geq 1$, an online learning rule $g_{\leq T_\eta}$ such that for any $\boldsymbol{y} := (y_t)_{t=1}^{T_\eta}$ of values in $\mathcal{Y}$ and any value $y \in \mathcal{Y}$, we have*

$$\frac{1}{T_\eta} \mathbb{E}\left[\sum_{t=1}^{T_\eta} \ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)\right] \leq \eta.$$

While this property may be quite difficult to interpret, it is satisfied by most reasonable value spaces that would be considered for regression. In particular, any totally-bounded space $(\mathcal{Y}, |\cdot|)$ satisfies F-TiME. Some non-totally-bounded spaces may also satisfy it, with the notable example of classification with a countably infinite number of classes and the standard $0-1$ loss: $(\mathcal{Y}, \ell) := (\mathbb{N}, \ell_{01})$.

We can now present the characterization for SOLAR. As in the realizable case, there is also always an optimistically universal learning rule, although it is in general much more complicated than 2C1NN.

Theorem 2.4 (Adversarial regression [BJ23])**.** *Fix a separable metric space $(\mathcal{X}, \rho)$ and a separable generalized-metric space $(\mathcal{Y}, \ell)$ such that the loss ℓ is bounded.*

- *If $(\mathcal{Y}, \ell)$ satisfies F-TiME. Then, SOLAR = SMV.*
- *If $(\mathcal{Y}, \ell)$ does not satisfy F-TiME. Then, SOLAR = CS.*

Further, in all cases there is an optimistically universal learning rule.

We do not cover the unbounded loss case here: we saw in the noiseless case that it was already very restrictive, this is a fortiori true for adversarial regression. As it turns out, in some cases, even under FS processes universal learning for adversarial responses may be impossible. The alternative is indeed either SOLAR = FS or SOLAR = $\emptyset$. We refer to Chapter 4 for further details.

Learning with unbounded losses with integrability responses. Although we mainly presented negative results for the unbounded loss case, we can recover positive results from the bounded loss case by adding assumptions on the form of the rewards. We propose the following *empirical integrability* condition on the rewards.

Definition 2.7 (Empirical integrability)**.** *A process $\mathbb{Y} = (Y_t)_{t\geq 1}$ is* empirically integrable *if there exists $y_0 \in \mathcal{Y}$ such that for any $\epsilon > 0$, almost surely there exists $M \geq 0$ for which*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, Y_t) \mathbb{1}_{\ell(y_0,Y_t)\geq M} \leq \epsilon.$$

In some sense, this is a necessary assumption for learning in unbounded value spaces. For instance, if the process $\mathbb{Y}$ was i.i.d., this exactly asks that $\ell(y_0, Y_1)$ has finite expectation which is somewhat necessary to learn in standard supervised learning settings. Also, if ℓ is bounded, the response process $\mathbb{Y}$ is automatically empirically integrable.

If we restrict the adversary to select response processes that are empirically integrable, we can define universally learnable processes in the same way as before and recover all results from bounded losses in Theorem 2.4.

At this point, we have a relatively complete picture of the universal learning landscape for full-feedback settings in which the observations O_t coincide exactly with the true values Y_t. These results are summarized in Table 2.1.

Learning setting	Bounded loss	Unbounded loss	Unbounded loss with empirically integrable responses
Noiseless responses	SOUL = SMV	SOUL = FS	Identical to bounded loss
Adversarial (or arbitrary) responses	Does $(\mathcal{Y}, \ell)$ satisfy F-TiME? $\begin{cases} \text{Yes} & \text{SOLAR} = \text{SMV} \\ \text{No} & \text{SOLAR} = \text{CS} \end{cases}$	Is ME achievable? $\begin{cases} \text{Yes} & \text{SOLAR} = \text{FS} \\ \text{No} & \text{SOLAR} = \emptyset \end{cases}$	Identical to bounded loss

Table 2.1: Universally learnable processes for full-feedback online learning (ME = Mean Estimation, see Chapter 4).

2.2.4 Contextual bandits with stationary rewards

We now switch gears and turn to partial-feedback learning settings, in which the learner does not observe the true value at each iteration. In particular, we focus on the contextual bandit problem is one of the core problems in this area of sequential decision-making. The main difference with the sequential prediction problem that we considered in the previous sections is that the only feedback is the reward of the selected prediction. In the bandit context, it is more common to talk about selected actions rather than selected prediction, hence we will use this terminology from now on and write $\mathcal{A}$ instead of $\mathcal{Y}$ for the action space.

The formal setup is as follows: a learner iteratively observes a context $X_t \in \mathcal{X}$, selects an action $a_t \in \mathcal{A}$, then receives a potentially stochastic reward $r_t \in [0, \infty)$ that depends on the context and selected action. This falls into the framework that was introduced in Section 2.2.1, where the observation is precisely the reward $O_t = r_t$ and the observation space is $\mathcal{O} = [0, \infty)$. This key difference that the learner only observes the reward of their action has some important practical and theoretical implications. On the practical side, it typically allows us to model a broader range of applications: for instance in clinical trials, one aims to learn the optimal personalized treatments but at each trial, one only gathers information about the treatment that was prescribed to the patient. On the theoretical side, partial feedback typically induces a fundamental trade-off between *exploration* and *exploitation*: it is sometimes beneficial to explore certain actions in the hope that they yield high rewards, rather than simply selecting the action that historically performed best.

Reward generating process. The classical model assumption for the rewards is that there is some underlying time-invariant conditional distribution that generates the rewards r_t conditionally on the selected action a_t and the context X_t for $t \geq 1$. Precisely, we assume that the rewards r_t for $t \geq 1$ are independent and follow a common conditional distribution $r_t \mid a_t, x_t \sim P_{r|a,X}$. For convenience, we let $\bar{r}(a, x) := \mathbb{E}_{r|P_{r|a,x}}[r \mid a, x]$ denote the immediate expected reward. As in the full-feedback setting, the main case of interest is when rewards are bounded and we will therefore assume that the rewards fall in $[0, 1]$.

Universal consistency. In this contextual bandit setting, our goal is to ensure that the algorithm converges to optimal actions given the contexts in order to maximize the rewards. Intuitively, such an optimal policy selects for the context $x \in \mathcal{X}$ an action in

$\arg\max_{a\in\mathcal{A}} \bar{r}(a,x)$. In practice, if the action space $\mathcal{A}$ is infinite, this may not be well-defined. Instead, the natural objective is to have algorithms that have low excess loss compared to any fixed measurable policy $f : \mathcal{X} \to \mathcal{A}$, which exactly matches the consistency definition Definition 2.3. Universally consistent algorithms are then naturally defined as follows.

Definition 2.8 (Universal consistency (contextual bandits))**.** *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$ and f. a learning rule. We say that f. is* universally consistent *under $\mathbb{X}$ for stationary contextual bandits if for any conditional distribution $P_{r|x,a}$ of the rewards, f. is consistent under $\mathbb{X}$ and rewards generated via $P_{r|x,a}$.*

The class of universally learnable processes SOCB (Strong Online Contextual Bandits) are then defined as

$$\text{SOCB} := \{\text{processes } \mathbb{X} : \exists \text{ learning rule } f. \text{ universally consistent under } \mathbb{X}\},$$

and optimistically universal learning rules are defined as in Definition 2.5 by replacing SOUL with SOCB.

While in the full-feedback case, previous works showed that i.i.d. processes and even stationary-ergodic were universally learnable, we were not aware of previous works that showed that even i.i.d. processes were universally learnable for contextual bandits with say finite action sets $\mathcal{A}$. Fortunately, we can still give precise characterizations of SOCB. This time, the characterization of learnable processes SOCB exhibits a trichotomy that depends on the action space $\mathcal{A}$: whether it is finite, countably infinite, or uncountable. The main result is the following.

Theorem 2.5 (Contextual bandits [BHJ22])**.** *Fix a separable metric space $(\mathcal{X},\rho)$ and a separable metrizable action space $\mathcal{A}$. For bounded rewards,*

- *If $\mathcal{A}$ is finite and $|\mathcal{A}| \geq 2$, then SOCB = SMV.*
- *If $\mathcal{A}$ is countably infinite, then SOCB = CS.*
- *If $\mathcal{A}$ is an uncountable separable metrizable Borel space, then SOCB = $\emptyset$.*

Further, in all cases, there is an optimistically universal learning rule.

Note that if the action space is finite, which corresponds to the classical contextual bandit setting, then we can still universally learn under all SMV processes. In particular, going from noiseless responses in the full-feedback to the partial-feedback of contextual bandits came at no cost for universal learning. On the other hand, if the action space is infinite, there is a big gap between the full-feedback case in which learning under CS processes was *always* possible, and the partial-feedback case in which universal learning is *never* possible. Indeed, because the reward is partial, the algorithm needs to explore for good actions but if these are uncountable, it may not even have time to explore all of these.

As should be expected, this "curse of exploration" can be significantly alleviated if we add some continuity assumptions to the contextual bandit model. A convenient condition is *uniform-continuity*, defined below. The metric on $\mathcal{A}$ is denoted by d.

Definition 2.9 (Uniformly-continuous rewards)**.** *We say that the reward mechanism* $r \sim P_{r|a,x}$ *is* uniformly-continuous *if for any* $\epsilon > 0$ *there exists* $\Delta(\epsilon) > 0$ *with*

$$\forall x \in \mathcal{X}, \forall a, a' \in \mathcal{A}, \quad d(a, a') \leq \Delta(\epsilon) \Rightarrow |\bar{r}(a, x) - \bar{r}(a', x)| \leq \epsilon.$$

Under this uniform continuity assumption, we can recover universal learning under the very large classes of processes CS and SMV. This is summarized in the following result which shows that the dichotomy becomes whether $(\mathcal{A}, d)$ is totally-bounded or not.

Theorem 2.6 (Contextual bandits with uniformly-continuous bounded rewards [BHJ22])**.** *Fix a separable metric space* $(\mathcal{X}, \rho)$ *and a metric action space* $(\mathcal{A}, d)$*. For bounded rewards,*

- *If* $\mathcal{A}$ *is totally-bounded and* $|\mathcal{A}| \geq 2$*, then* $\textsc{Socb} = \textsc{SMV}$*.*
- *If* $\mathcal{A}$ *is non-totally-bounded, then* $\textsc{Socb} = \textsc{CS}$*.*

Further, in all cases, there is an optimistically universal learning rule.

2.2.5 Adversarial contextual bandits

The major assumption in the previous contextual bandit setting was that the rewards were generated according to some underlying conditional distribution $P_{r|a,x}$. While this is the most standard assumption in the contextual bandit literature, it is well understood that in practical implementations, reward mechanisms can evolve over time, potentially adversarially and depending on the learner's actions. In this section, we investigate the much more general adversarial rewards in contextual bandits. This should be interpreted as a first significant step towards more general learning frameworks such as reinforcement learning where the evolution of the reward mechanism depends on the past history. Unlike any of the settings we previously considered, we will see that giving optimistically universal algorithms for adversarial contextual bandits is in fact *impossible*. However, universal learning is still in general achievable for general processes beyond i.i.d. and stationary ergodic processes.

Before diving into more details, we first need to properly define the non-stationary reward models. In addition to stationary rewards, many choices are possible and we only highlight three standard non-stationarity models. *Memoryless* rewards are the mildest form of generalization from stationary rewards: these are non-adaptive and simply shift distributions over time. On the other hand, *oblivious* rewards may depend on the specific sequence of observed contexts $\boldsymbol{X}_{\leq t}$, while *online* rewards are fully adaptive: these may additionally depend on the specific actions and rewards of the algorithm. These are formally defined below.

Definition 2.10 (Reward models)**.** *The reward mechanism is said to be*

- stationary *if there is a conditional distribution* $P_{r|a,x}$ *such that the rewards* $(r_t)_{t\geq 1}$ *given their selected action* a_t *and context* X_t *are independent and follow* $P_{r|a,x}$
- memoryless *if there are conditional distributions* $(P_{r|a,x,t})_{t\geq 1}$ *such that* $(r_t)_{t\geq 1}$ *given their selected action* a_t *and context* X_t *are independent for* $t \geq 1$ *and respectively follow* $P_{r|a,x,t}$

- oblivious *if there are conditional distributions* $(P_{r|a,\boldsymbol{x}_{\leq t}})_{t\geq 1}$ *such that* r_t *given the selected action* a_t *and the past contexts* $\mathbb{X}_{\leq t}$*, follows* $P_{r|a,\boldsymbol{x}_{\leq t}}$
- online *if there are conditional distributions* $(P_{r|\boldsymbol{a}_{\leq t},\boldsymbol{x}_{\leq t},\boldsymbol{r}_{\leq t-1}})_{t\geq 1}$ *such that* r_t *given the sequence of selected actions* $\boldsymbol{a}_{\leq t}$ *and the sequence of contexts* $\mathbb{X}_{\leq t}$ *and received rewards* $\boldsymbol{r}_{\leq t-1}$*, follows* $P_{r|\boldsymbol{a}_{\leq t},\boldsymbol{x}_{\leq t},\boldsymbol{r}_{\leq t-1}}$.

Having fixed the reward model, as in the case of stationary contextual bandits (Definition 2.8), we can define the notion of universal consistency under some process $\mathbb{X}$ for each reward model. A learning rule is universally consistent under $\mathbb{X}$ if for any form of rewards within the specified reward model, it is consistent under $\mathbb{X}$ and for these rewards as per Definition 2.3. We then denote by $\textsc{Soab}_{model}$ (Strong Online Adversarial contextual Bandits) the set of universally learnable processes for adversarial contextual bandits within a specified model $model \in \{stationary, memoryless, oblivious, online\}$:

$$\textsc{Soab}_{model} := \{\text{processes } \mathbb{X} : \exists \text{ learning rule } f. \text{ universally consistent for } model \text{ rewards under } \mathbb{X}\}.$$

For instance, for stationary rewards, we have exactly $\textsc{Soab}_{stationary} = \textsc{Socb}$, which we covered in the previous section. Of course, the more general the reward, model, the harder the universal learning objective becomes. Hence, we have

$$\textsc{Soab}_{online} \subseteq \textsc{Soab}_{oblivious} \subseteq \textsc{Soab}_{memoryless} \subseteq \textsc{Soab}_{stationary}.$$

Our main result for adversarial contextual bandits is that in the main case of interest when the action set $\mathcal{A}$ is finite, for generic context spaces $\mathcal{X}$, there never exists an optimistically universal learning rule. This holds whenever $\mathcal{X}$ admits some non-atomic probability measure, which includes for instance any uncountable Polish space. The other cases when $\mathcal{A}$ is infinite are somewhat closer to the case of stationary rewards, in fact under the strongest form of online rewards, we can still universally learn under the same class of stochastic processes $\mathbb{X}$.

Theorem 2.7 (Adversarial contextual bandits)**.** *Fix a separable metric space* $(\mathcal{X}, \rho)$ *and a metrizable action space* $\mathcal{A}$*. For bounded rewards,*

1. *If* $\mathcal{A}$ *is finite and* $|\mathcal{A}| \geq 2$*,*
   - *If* $\mathcal{X}$ *admits a non-atomic probability measure, there do not exist optimistically universal learning rules for any non-stationary reward model from Definition 2.10.*
   - *Otherwise, there exists an optimistically universal learning rule for all reward models from Definition 2.10 and* $\textsc{Soab}_{online} = \textsc{Soab}_{stationary} = SMV$*.*
2. *If* $\mathcal{A}$ *is countably infinite, there exists an optimistically universal learning rule for all reward models from Definition 2.10 and* $\textsc{Soab}_{online} = \textsc{Soab}_{stationary} = CS$*.*
3. *Let* $\mathcal{A}$ *be an uncountable separable metrizable Borel space, then universal learning is never achievable and* $\textsc{Soab}_{online} = \textsc{Soab}_{stationary} = \emptyset$*.*

Learning setting	Unrestricted rewards	Uniformly-continuous rewards
Bounded rewards	Finite $\mathcal{A}$: SMV Countably infinite $\mathcal{A}$: CS Uncountable $\mathcal{A}$: $\emptyset$	Totally-bounded $\mathcal{A}$: SMV Non-totally-bounded $\mathcal{A}$: CS
Unbounded rewards	Countable $\mathcal{A}$: FS Uncountable $\mathcal{A}$: $\emptyset$	FS

Table 2.2: Universally learnable processes for stationary contextual bandits

Learning setting	Unrestricted rewards	Uniformly-continuous rewards
Bounded rewards	Finite $\mathcal{A}$, $\mathcal{X}$ with non-atomic proba. measure: $\text{CS} \subsetneq \mathcal{C} \subsetneq \text{SMV}$ Finite $\mathcal{A}$, $\mathcal{X}$ without non-atomic proba. measure: SMV Countably infinite $\mathcal{A}$: CS Uncountable $\mathcal{A}$: $\emptyset$	Totally-bounded $\mathcal{A}$: same as for bounded rewards and finite $\mathcal{A}$ Non-totally-bounded $\mathcal{A}$: CS
Unbounded rewards	Countable $\mathcal{A}$: FS Uncountable $\mathcal{A}$: $\emptyset$	FS

Table 2.3: Universally learnable processes $\mathcal{C}$ for adversarial contextual bandits

We focus on action spaces $\mathcal{A}$ and suppose that $\mathcal{X}$ has a non-atomic probability measure, which is arguably the most interesting case for adversarial contextual bandits. Although the previous results show that *optimistic* learning is impossible, this does not mean that *universal* learning is impossible. We can show that one can still achieve universal consistency under very general classes of processes (in fact, beyond CS processes). Instead, the above impossibility result implies that a learner needs more initial information about the process $\mathbb{X}$ to achieve universal consistency: the optimist's assumption alone is not sufficient. At a very high level, a learner cannot distinguish between using a strategy at the population level (using all historical data) or at the individual level (using historical data only from instances very similar to the current context X_t).

Characterizing the exact class of universally learnable process for adversarial rewards SOAB in this case turns out to be quite challenging. In fact, for non-stationary reward models, the class strictly lies somewhere in between CS and SMV. We give a full characterization of SOAB_{online} for fully adaptive rewards in Chapter 6 as well as necessary and sufficient conditions for the other models, but the corresponding classes of processes significantly depart from the definition of CS of SMV and we will not give detailed results here for the sake of simplicity.

This concludes our overview of the universal results for contextual bandits. These are summarized in Tables 2.2 and 2.3. For completeness, we also added all results corresponding to unbounded rewards with and without uniform-continuity assumptions. As in the stationary contextual bandit case, uniform-continuity allows the recovery of all the positive results from the bounded reward case. Studying universal learning for other more complex partial-feedback machine learning settings is certainly possible. In fact, all results can be directly

lifted to finite-horizon episodic Reinforcement Learning (RL) in which a learner evolves in a finite-horizon Markov decision process at each iteration, by viewing the RL problem as a contextual bandit problem on a larger state and action space.

Chapter 3

Universal Realizable Online Learning

3.1 Introduction

In this chapter, we study universal learning in the realizable setting, as defined in the overview Chapter 2. We consider the fundamental question of learnability and generalizability for online learning. In this framework, a learner is sequentially given input points $\mathbb{X} := (X_t)_{t\geq 0}$ from a general separable metric *instance space* $(\mathcal{X}, \rho)$ and observes the corresponding values $\mathbb{Y} := (Y_t)_{t\geq 0}$ from a separable near-metric *value space* $(\mathcal{Y}, \ell)$. The learner's goal is to predict the values before their observation. The input points are given according to some stochastic process $\mathbb{X}$ on $\mathcal{X}$ and we assume that the process $\mathbb{Y}$ is generated from $\mathbb{X}$ in a *noiseless* fashion, i.e., that there exists an unknown measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ such that $Y_t = f^*(X_t)$ for all $t \geq 0$. At time step t, the learner outputs a prediction $\hat{Y}_t$ based solely on the historical data $(X_u, Y_u)_{u<t}$ and the new input point X_t. We wish to obtain low long-run average errors $\frac{1}{t}\sum_{u\leq t} \ell(Y_u, \hat{Y}_u)$. Specifically, we consider two types of consistency: strong consistency is achieved when the average error converges to 0 almost surely; and weak consistency is achieved when the expected average error converges to 0. We are interested in *universal* online learning, in which we ask for consistency for any unknown measurable target function f^*. In this framework, the two main questions are (1), to characterize the input processes $\mathbb{X}$ for which universal consistency is achievable, and (2), if possible, provide a learning rule which would guarantee universal consistency whenever such objective is achievable. For a detailed discussion of the general motivation for universal learning and related work, we refer to Section 2.1.

3.1.1 Contributions

In this chapter, we settle these universal learning questions. Of particular interest, we show that there exist, and explicitly provide optimistically universal learning rules for general online learning. Further, in the more interesting case of *bounded losses*, universal learning can be achieved under very large classes of processes, well beyond say i.i.d. or stationary ergodic processes, which significantly generalized prior works. The contributions can be summarized as follows.

- For bounded losses, we propose a class of learning rule kC1NN for $k \geq 2$, which

we prove are strongly and weakly optimistically universal for general separable metric instance spaces $(\mathcal{X}, \rho)$ and separable near-metric value spaces $(\mathcal{Y}, \ell)$ with bounded loss. These learning rules are simple variants of the classical 1-nearest neighbor (1NN). They essentially perform 1NN on a restricted dataset by deleting any input point from the historical dataset whenever it has been used as nearest neighbor at least k times. We also show that any $(k_t)_t-$nearest neighbor fails to be optimistically universal under very mild conditions on the sequence $(k_t)_t$ even for very simple input spaces $\mathcal{X}$ e.g. Euclidean spaces. Finally, we give a complete characterization of processes admitting strong and weak universal learning. This closes the questions on universal online learning stated as open problems in [Han21b].

- For unbounded losses, we show that the only learnable processes are those that almost surely contain only a finite number of distinct elements. As a result, a simple memorization algorithm suffices to be optimistically universal, which simply remembers all past data points (X_s, Y_s), $s < t$, and if the new X_t satisfies $X_t = X_s$ for some $s < t$, it predicts Y_s. If $\mathbb{X}$ has only a finite number of distinct elements, then clearly this strategy has only finitely many non-zero losses, and hence would be universally consistent. This result is rather negative, however, since it implies that universal learning with unbounded losses is quite restrictive.

3.1.2 Organization of the chapter.

The rest of this chapter is organized as follows. In Section 3.2, we recall the universal learning formal setup and present the two main questions of this topic. The main results are then stated in Section 3.3. We next first focus on bounded losses. In Section 3.4 we focus on nearest neighbor learning rules and show that they are not universally consistent, by constructing learnable processes for which nearest neighbor methods fail. This example gives motivation for the 2C1NN learning rule, constructed in Section 3.5, and then show that it is optimistically universal for binary classification and in the simpler case of $\mathcal{X} = [0, 1]$. The generalization to any separable metrizable context space is done in Section 3.6. Last, to go from binary classification to any general value space, we design a general-purpose reduction in Section 3.7. Altogether, this provides a complete characterization of the set of learnable processes for strong universal learning with bounded losses. We then turn to weak universal learning in Section 3.8. We then turn to unbounded losses and prove our results in Section 3.9, in this case, strong and weak learning coincide.

3.2 Formal Setup and Preliminaries

Instance and value space. In this chapter, we follow the general framework of online learning where one observes an input sequence $\mathbb{X} = (X_t)_{t\geq 1}$ of points in a separable metric *instance space* $(\mathcal{X}, \rho)$, together with their corresponding target values $\mathbb{Y} = (Y_t)_{t\geq 1}$ coming from a separable near-metric *value space* $(\mathcal{Y}, \ell)$. We recall from Definition 2.1 that the loss $\ell : \mathcal{Y}^2 \to [0, \infty)$ is said to be a near-metric if it is symmetric $\ell(y_1, y_2) = \ell(y_2, y_1)$, satisfies $\ell(y_1, y_2) = 0$ if and only if $y_1 = y_2$, and also satisfies a relaxed triangle inequality

$\forall y_1, y_2, y_3 \in \mathcal{Y}^3 : \ell(y_1, y_3) \leq c_\ell(\ell(y_2, y_1) + \ell(y_2, y_3))$, where c_ℓ is a fixed constant. Note that all metrics are near-metrics with $c_\ell = 1$. As an important example for regression, the squared loss is near-metric with $c_\ell = 2$. We denote by $\bar{\ell} := \sup_{y_1, y_2 \in \mathcal{Y}} \ell(y_1, y_2)$ the loss function supremum.

Input and output processes. In an effort to study non-i.i.d. processes, the input sequence of points is a general stochastic process on the Borel space $(\mathcal{X}, \mathcal{B})$ induced by a metric ρ. This is a major difference with a majority of the relevant statistical learning literature imposing ad-hoc hypothesis on $\mathbb{X}$. We consider a noiseless setting in which the output values $\mathbb{Y}$ are generated from $\mathbb{X}$ through an unknown measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ such that $Y_t = f^*(X_t)$ for all $t \geq 1$.

Online learning and consistency. In *online* learning, the learning process is sequential: at time $t \geq 1$, one observes a new input data-point X_t and outputs a prediction $\hat{Y}_t$ based solely on the historical data $(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1})$ and the new covariate X_t. We measure the performance of the learning rule through the loss function ℓ. *Strong* consistency is achieved when the algorithm obtains asymptotic average loss 0 almost surely. Alternatively, a learning rule is *weakly* consistent when it guarantees 0 asymptotic average loss in expectation. We now formally write these notions. A learning rule is a sequence $f_\cdot = \{f_t\}_{t=1}^\infty$ of measurable functions with $f_1 : \mathcal{X} \to \mathcal{Y}$ and $f_t : \mathcal{X}^{t-1} \times \mathcal{Y}^{t-1} \times \mathcal{X} \to \mathcal{Y}$ for $t \geq 2$. Given a history $(X_i, Y_i)_{i<t}$ and a new input point X_t, the rule $f_\cdot$ makes the prediction $f_t(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t)$ for Y_t and $t \geq 2$. For simplicity, for $t = 1$ we may also use the notation $f_1(\mathbb{X}_{<1}, \mathbb{Y}_{<1}, X_1)$ instead of $f_1(X_t)$. As an important example, the *memorization learning rule* $\{f_t\}_{t=1}^\infty$ is defined as follows:

$$f_t((x_i)_{i<t}, (y_i)_{i<t}, x_t) = \begin{cases} y_i & \text{if } x_t = x_i, \\ y_0 & \text{if } x_t \notin \{x_i\}_{i<t}, \end{cases}$$

where $y_0 \in \mathcal{Y}$ is some arbitrary default response. We write the average loss at time T as

$$\mathcal{L}_\mathbb{X}(f_\cdot, f^*; T) := \frac{1}{T} \sum_{t=1}^{T} \ell(f_t(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t), f^*(X_t)).$$

We aim to minimize the long-run average loss. The online learning rule $f_\cdot$ is strongly consistent under the input process $\mathbb{X}$ and for the target function f^* when $\mathcal{L}_\mathbb{X}(f_\cdot, f^*; T) \to 0 \ \ (a.s.)$. For simplicity, we define $\mathcal{L}_\mathbb{X}(f_\cdot, f^*) = \limsup_{T \to \infty} \mathcal{L}_\mathbb{X}(f_\cdot, f^*; T)$. Therefore, the above condition can be rewritten as $\mathcal{L}_\mathbb{X}(f_\cdot, f^*) = 0 \ \ (a.s.)$. We also consider weak learning: similarly, $f_\cdot$ is weakly consistent under $\mathbb{X}$ and for f^* when $\mathbb{E}\mathcal{L}_\mathbb{X}(f_\cdot, f^*; T) \to 0$.

Universal consistency and optimistically universal learning rules. Following the work of [Han21a], we are interested in learning rules that achieve strong (resp. weak) consistency under a specific input sequence $\mathbb{X}$ for all measurable target functions $f^* : \mathcal{X} \to \mathcal{Y}$. Such learning rules are said to be strongly (resp. weakly) *universally consistent* under $\mathbb{X}$. We define SOUL as the set of all stochastic processes $\mathbb{X}$ for which strong universal online learning is achievable by some learning rule. Similarly, we denote by WOUL the set of all processes $\mathbb{X}$ that admit weak universal online learning. These sets may depend on the setup

$(\mathcal{X}, \rho), (\mathcal{Y}, \ell)$ so we will specify $\textsc{Soul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)}$ and $\textsc{Woul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)}$ when the spaces are not clear from the context. In this framework, two main areas of research are (1) characterizing the sets Soul (resp. Woul) for a given setup in terms of the properties of the stochastic process $\mathbb{X}$, and (2) identifying learning rules which are strongly (resp. weakly) universally consistent for any input process $\mathbb{X}$ in Soul (resp. Woul), i.e. that achieve strong (resp. weak) universal consistency whenever it is possible. These are called *optimistically universal* learning rules.

Prior work for bounded losses. In the present chapter we first focus on the *bounded* loss case i.e. $\bar{\ell} < \infty$, for which both questions prior to this work. Further, this is the main case of interest for universal online learning since contrary to the unbounded case, for bounded losses, it is known that the set of learnable processes Soul contains in particular all i.i.d. processes [Han21a]. In fact, the simple 1-nearest neighbor (1NN) learning rule achieves strong (and weak) universal consistency for all i.i.d. processes $\mathbb{X}$ in for the Euclidean space $\mathbb{X} = \mathbb{R}^d$ [Dev+94]. It is even known that the $(k_t)_t$-neighbor algorithm ($(k_t)_t$NN) with $k_t / \log t \to \infty$ and $k_t/t \to 0$ achieves Bayes minimal risk in the noisy setting for large classes of input spaces $\mathcal{X}$ [CD14]. This implies in particular that kNN achieves strong universal consistency in our noiseless setting for these input spaces. However, it was an open question whether there exist simple input spaces $\mathcal{X}$—e.g. Euclidean spaces—for which some kNN algorithms would be optimistically universal. In other terms, does there exist an input process $\mathbb{X}$ such that 1NN fails to achieve consistency for some target function f^* but universal consistency would still be achieved by some other—more sophisticated—learning rule? No characterization of Soul was known either, although [Han21a] proposed a necessary condition for belonging to Soul and conjectured that it is also sufficient. We refer to this condition as SMV (Sublinear Measurable Visits). Intuitively, it asks that for any measurable partition of the input space $\mathcal{X}$, the process $\mathbb{X}$ only visits a sublinear number of its regions. Note that this condition does not depend on the choice of output setup $(\mathcal{Y}, \ell)$. We recall its definition from Condition SMV.

Condition SMV. *For every disjoint sequence $\{A_k\}_{k=1}^{\infty}$ of measurable sets of $\mathcal{X}$ such that $\bigcup_{k=1}^{\infty} A_k = \mathcal{X}$, (every countable measurable partition),*

$$|\{k \geq 1 : A_k \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T), \quad (a.s.).$$

By abuse of notation, let SMV be the collection of processes $\mathbb{X}$ satisfying this condition.

For the weak setting we can define a similar condition WSMV (weak sub-linear measurable visits).

Condition WSMV. *For every disjoint sequence $\{A_k\}_{k=1}^{\infty}$ of measurable sets of $\mathcal{X}$ with $\bigcup_{k=1}^{\infty} A_k = \mathcal{X}$, (every countable measurable partition),*

$$\mathbb{E}[|\{k \in \mathbb{N} : A_k \cap \mathbb{X}_{<T} \neq \emptyset\}|] = o(T).$$

By abuse of notation, let WSMV be the collection of processes $\mathbb{X}$ satisfying this condition.

[Han21a] showed that these conditions are necessary for strong and weak universal learning.

Proposition 3.1 ([Han21a]). *For any separable Borel space $\mathcal{X}$ and separable near-metric output setting $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$ we have $\textsc{Soul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} \subset \textit{SMV}_{(\mathcal{X},\rho)}$ and we have that $\textsc{Woul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} \subset \textit{WSMV}_{(\mathcal{X},\rho)}$.*

The intuition should be rather clear: whenever the process $\mathbb{X}$ explores for the first time a new region A_k, an algorithm has no prior information about the true value on this region, hence will incur at least an error $\frac{1}{2}$ in expectation. If the conditions SMV (resp. WSMV) are not satisfied by $\mathbb{X}$, then the corresponding rate of exploration will by hypothesis be linear, which means that the algorithm makes a linear number of mistakes and is as a result, not universally consistent.

However, it was an open question whether SMV (resp. WSMV) is also a sufficient condition for strong (resp. weak) universal learning. Together with the question of the existence of an optimistically universal learning rule, these are the main objectives for universal online learning. These questions are posed in the COLT 2021 open problems [Han21b], which we now formally restate.

Question 3.1 ([Han21b]). *Does there exist an optimistically universal online learning algorithm? (in either the weak or strong sense)*

Question 3.2 ([Han21b]). *Is SMV (resp. WSMV) equal to the set of all $\mathbb{X}$ such that strong (resp. weak) universal online learning is possible under $\mathbb{X}$?*

It is important to note that these questions are easily solved in the case where $\mathcal{X}$ is countable [Han21a]. Therefore, the main interest is to answer these questions for *any* uncountable $\mathcal{X}$. In fact, [Han21b] even announced a \$5000 (resp. \$1000) reward for solving Question 3.1 (resp. Question 3.2) for the Euclidean $\mathcal{X} = \mathbb{R}^d$ case. Both questions will be solved in Section 3.5.1 for $\mathcal{X} = [0, 1]$ specifically. This is a rather general case because its extension to all standard Borel spaces $\mathcal{X}$ is immediate through an equivalence result from Kuratowski of all uncountable standard Borel spaces. For instance, this solves the question for all Euclidean spaces $\mathbb{R}^d$ for $d \geq 1$. Most importantly, the special case $\mathcal{X} = [0, 1]$ allows for a simplified exposition and provides all useful intuitions. The complete result holds for all separable Borel spaces and is presented in Section 3.6.

Prior work for unbounded losses. In the case of *unbounded* losses ℓ, the existence of an optimistically universal online learning rule was settled by [Han21a].

This work also expresses a condition which characterizes the family of processes $\mathbb{X}$ that admit the existence of universally consistent online learning rules for any (and all) unbounded losses. However, the definition of the optimistically universal learning rule given in that work, the proof that it satisfies this property, and also the proofs establishing that the proposed condition indeed characterizes the relevant family of processes, are actually quite complex. For instance, the learning rule involves identifying a function contained in a certain countable function class $\tilde{\mathcal{F}}$, satisfying constraints on its losses relative to various other values, and the proof proceeds via arguing that there exists a choice of $\tilde{\mathcal{F}}$ that is *dense* in the set of all measurable functions, in a sense relevant to learning under every $\mathbb{X}$ satisfying the condition. However, [Han21a] also poses an interesting open question regarding a potential dramatic simplification of this theory. The essential question is the following:

Question 3.3 ([Han21b]). *For unbounded losses, is it true that there exist universally consistent online learning rules under $\mathbb{X}$ if and only if $\mathbb{X}$ almost surely has a finite number of distinct elements?*

We refer to the above simple condition as the *Finite Support* (FS) condition, which we recall the definition from Condition FS.

Condition FS. *The process $\mathbb{X}$ satisfies $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X} \neq \emptyset\}| < \infty \quad (a.s.)$. By abuse of notation let FS be the collection of processes satisfying this condition.*

Notations. For any sequence $\boldsymbol{x}$, we will use the following notations when analyzing finite time horizons: $\boldsymbol{x}_{\leq t} := \{x_1, ..., x_t\}$ and $\boldsymbol{x}_{<t} := \{x_1, ..., x_{t-1}\}$ for simplicity. For a metric space $(\mathcal{X}, \rho)$, a point $x \in \mathcal{X}$ and $r \geq 0$, we denote by $B_\rho(x, r) := \{x' \in \mathcal{X},\ \rho(x, x') < r\}$ the open ball centered in x of radius r, and $S_\rho(x, r) = \{x' \in \mathcal{X},\ \rho(x, x') = r\}$ the sphere centered in x of radius r. We might omit the metric ρ in subscript if there is no ambiguity. We also denote by ℓ_{01} the indicator loss function, i.e., $\ell_{01}(i, j) = \mathbb{1}(i \neq j)$. Since it is a metric, it is also a near-metric with $c_\ell = 1$. For simplicity, we will use the same notation ℓ_{01} irrespective of the output space $\mathcal{Y}$. For any measurable set A, we denote by $\mathbb{1}_A$ the function $\mathbb{1}_A(\cdot) := \mathbb{1}_{\cdot \in A}$. We will denote by $|\cdot|$ any norm on $\mathbb{R}$. Recall that all norms are equivalent on finite dimensional spaces, hence the topology induced by these metrics is identical. When the space $(\mathcal{X}, \rho)$ is obvious from the context, we may reduce the notation $\text{SOUL}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)}$ to $\text{SOUL}_{(\mathcal{Y},\ell)}$. We may also omit the loss ℓ when there is no ambiguity.

3.3 Main Results

We start with the case of bounded losses. We first show that the simple nearest neighbor rule (1NN) is not optimistically universal. The proof generalizes to general $(k_t)_t$-nearest neighbor algorithms under very mild assumptions on $(k_t)_t$.

Theorem 3.1. *The $(k_t)_t-$nearest neighbor learning rule is not strongly optimistically universal for the input space $\mathcal{X} = [0, 1]$ with usual topology and for binary classification, for any sequence $(k_t)_t$ such that $k_t = o\left(\frac{t}{(\log t)^{1+\delta}}\right)$ for any $\delta > 0$.*

This is obtained by constructing a specific process $\mathbb{X} \in \text{SOUL}_{([0,1],|\cdot|),(\{0,1\},\ell_{01})}$ under which nearest neighbor is not universally consistent. Intuitively, 1NN fails on the process because certain "bad" data points are used an arbitrarily large number of times as nearest neighbor for future input points and hence, induce a large number of mistakes for 1NN. To resolve this issue, we propose a new learning rule 2-Capped-1-Nearest-Neighbor (2C1NN), a variant of the classical 1NN, designed to ensure that the number of times each datapoint is used as nearest neighbor is capped by 2. Specifically, once a datapoint X_t has been used as nearest neighbor twice, it is deleted from the training dataset. We show that this is an optimistically universal learning rule for both strong universal learning and weak universal learning.

Theorem 3.2. *For any separable Borel space $\mathcal{X}$, and any separable near-metric output setting $(\mathcal{Y}, \ell)$ with bounded loss, i.e., $\sup_{y_1,y_2} \ell(y_1, y_2) < \infty$, 2C1NN is a strongly (resp. weakly) optimistically universal learning rule.*

More generally, we can define learning rules kC1NN for any $k \geq 2$. The proof further shows that all kC1NN is optimistically universal for any $k \geq 2$. Further, we give a characterization of the processes admitting strong and weak universal learning.

Theorem 3.3. *For any separable Borel space $\mathcal{X}$, and any separable near-metric output setting $(\mathcal{Y}, \ell)$ with $0 < \sup_{y_1,y_2} \ell(y_1, y_2) < \infty$, we have*

$$\textsc{Soul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} = SMV_{(\mathcal{X},\rho)} \quad \textit{and} \quad \textsc{Woul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} = WSMV_{(\mathcal{X},\rho)}.$$

If $\sup_{y_1,y_2} \ell(y_1, y_2) = 0$, then the loss is identically null. Therefore, all stochastic processes are strongly and weakly learnable.

It is worth noting that although the sets Soul and Woul differ—the set of weakly learnable processes Woul is larger than the set of strongly learnable processes Soul—the same learning rule 2C1NN is optimistically universal in both strong and weak settings. Theorem 3.2 and Theorem 3.3 close the two open questions of the existence of an optimistically universal learning rule and a characterization of the set of learnable input sequences, formulated in [Han21b].

We next turn to unbounded losses and answer the corresponding open question with the positive. Our main result for unbounded losses can then be stated as follows.

Theorem 3.4. *For $\mathcal{X}$ any separable metric space and ℓ any unbounded loss, Soul = FS. Further, the memorization learning rule is optimistically universal.*

Altogether, these results give a complete picture of universal learning in this realizable setting: we can always provide an optimistically universal learning rule and characterize the class of universally learnable processes: while these are quite restrictive for unbounded losses, in the more standard learning setting of bounded losses, we show that universal learning can be achieved well beyond say i.i.d. or stationary ergodic processes.

3.4 On Nearest Neighbor Consistency

Going back to the bounded loss case, a natural candidate for good learning rules in general spaces are the nearest neighbor algorithms. We recall that the $(k_t)_t-$nearest neighbor ($(k_t)_t$NN) learning rule, at step t, considers the closest k_t neighbors to the new input point and follows the majority vote to make its prediction. Indeed, in the Euclidean space, 1NN is universally consistent under all i.i.d. processes [Dev+94]. Further, $(k_t)_t$NN learning rules with $k_t/\log t \to \infty$ and $k_t/t \to 0$ are also universally consistent under i.i.d. processes for smooth classes of input spaces $\mathcal{X}$ [CD14]. However, it is known that there exist separable input spaces for which no $(k_t)_t$NN algorithm achieves universal consistency [CG06]. In this section, we show that $(k_t)_t$NN learning rules are not optimistically universal even on the interval $\mathcal{X} = [0, 1]$.

Theorem 3.1. *The $(k_t)_t-$nearest neighbor learning rule is not strongly optimistically universal for the input space $\mathcal{X} = [0, 1]$ with usual topology and for binary classification, for any sequence $(k_t)_t$ such that $k_t = o\left(\frac{t}{(\log t)^{1+\delta}}\right)$ for any $\delta > 0$.*

As a direct consequence, $(k_t)_t-$nearest neighbors are not optimistically universal for any input spaces $\mathcal{X}$ such that there exists a measurable injection $[0,1] \to \mathcal{X}$ and for any output setting $(\mathcal{Y}, \ell)$ with bounded loss and at least two distinct values $y_1, y_2 \in \mathcal{Y}$ such that $\ell(y_1, y_2) > 0$. In particular, this shows that $(k_t)_t$NN algorithms are not optimistically universal in Euclidean spaces. To prove Theorem 3.1, we first define the set of processes with convergent relative frequencies CRF as the set of processes $\mathbb{X}$ such that $\forall A \in \mathcal{B}$,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_A(X_t) \quad \text{exists (a.s.).}$$

We then explicitly construct a process $\mathbb{X}^{(1)} \in$ CRF on which $(k_t)_t-$nearest neighbor fails. Because convergent relative frequencies processes are learnable CRF $\subset$ Soul [Han21a], this shows that $(k_t)_t$NN is not optimistically universal for the online learning setting. Note that we have CRF $\subsetneq$ Soul for any infinite space $\mathcal{X}$. As a remark, it was already known that the self-adaptive/inductive nearest neighbor learning rule is not optimistically universal for the self-adaptive setting [Han21a] (Section 3.2). Inductive learning differs from online learning in that the learner has access to a fixed historical dataset $(X_t, Y_t)_{t<T}$ and from time T has to commit to a (non-adaptive) learning rule. Self-adaptive learning is an intermediate setting between inductive learning and online learning where the learner can be adaptive on observed instances $(X_t)_{t\geq T}$ but not the values $(Y_t)_{t\geq T}$. Hence, the self-adaptive/inductive nearest neighbor learning rule corresponds to performing nearest neighbor with the fixed dataset $(X_u, Y_u)_{u<T}$ for any $t \geq T$. The performance of this learning rule is taken as a double limit: first as $t \to \infty$, then as $T \to \infty$. We refer to [Han21a] for details on these settings. Similarly to the set Soul, we can define the set Sual of processes $\mathbb{X}$ admitting strong universal learning in the self-adaptive setting. The proof that self-adaptive nearest neighbor is not optimistically universal is also constructive but not relevant for the online setting because it relies on a completely different process $\mathbb{X}^{(2)} \in$ Sual under which self-adaptive 1-nearest neighbor fails but online learning 1-nearest neighbor is universally consistent. Indeed, the set of learnable processes for online learning is larger than the set of learnable processes for self-adaptive learning Sual $\subset$ Soul, and strictly larger whenever $\mathcal{X}$ is infinite [Han21a].

Proof of Theorem 3.1 Let $\delta > 0$ and a sequence $k_t = o\left(\frac{t}{(\log t)^{1+\delta}}\right)$. To show that $(k_t)_t-$NN is not optimistically universal, we construct a process $\mathbb{X} \in$ Soul on which $(k_t)_t-$NN has asymptotic error rate 1. We denote by $\mathcal{D}_p := \{\frac{i}{2^p}, 0 \leq i \leq 2^p,\ i \text{ odd}\}$ the set of dyadics of order p i.e. with reduced denominator 2^p, and $\mathcal{D}$ for the set of dyadics. Let $\epsilon > 0$ such that $\frac{1+2\epsilon}{1-2\epsilon} < 1 + \frac{\delta}{2}$. Then pose for $k \geq 1$,

$$n_k = \lfloor e^{k^{1/2-\epsilon}} \rfloor, \quad d_k = \min\left(\left\lfloor \frac{n_k}{(\log n_k)^{1+\delta}} \right\rfloor, n_{k+1} - n_k - 1\right), \quad p_k = 4^k.$$

First note that $n_{k+1} - n_k \sim \left(\frac{1}{2} - \epsilon\right) \frac{n_k}{k^{1/2+\epsilon}} \sim \left(\frac{1}{2} - \epsilon\right) \frac{n_k}{(\log n_k)^{\frac{1/2+\epsilon}{1/2-\epsilon}}}$ therefore we obtain

$$d_k = o\left(\frac{n_k}{(\log n_k)^{1+\delta/2}}\right) = o(n_{k+1} - n_k).$$

Also, for k large enough, $d_k = \left\lfloor \frac{n_k}{(\log n_k)^{1+\delta}} \right\rfloor$. We now construct a process $\mathbb{X}$ on $\mathcal{X}$. Let $(U_k)_{k\geq 1}$ be an i.i.d. sequence of uniforms $\mathcal{U}([0,1])$ and $(D_k)_{k\geq 1}$ a sequence of independent random variables—also independent of $(U_k)_k$—such that $D_k \sim \mathcal{U}(\mathcal{D}_{p_k})$. Additionally, we denote by $D_{k,i}$ the $i-$th closest dyadic of order p_k to D_k. For instance, $D_{k,1} = D_k$, and $|D_{k,i} - D_k| \leq \frac{i}{2^{p_k-1}}$. For intuition, if D_k is not close to the boundary of $[0,1]$, we have $D_{k,i} = D_k + (-1)^i \cdot \frac{\lfloor i/2 \rfloor}{2^{p_k}}$. We now define the process $\mathbb{X}$ as follows for any $k \geq 1$,

$$X_{n_k+i} = D_{k,i+1}, \quad 0 \leq i \leq d_k \quad \text{and} \quad X_{n_k+d_k+j} = D_k + \frac{U_k - D_k}{2^{n_k} 4^j}, \quad 1 \leq j < n_{k+1} - n_k - d_k.$$

We first prove that $(k_t)_t-$NN is not consistent for the function $f^* = 1_{\mathcal{D}}$. For any $k \geq 1$,

$$\mathbb{P}\left[\min_{t<n_k} |X_t - D_k| < \frac{1}{2^{n_k}}\right] \leq \sum_{t<n_k} \mathbb{P}\left[X_t - \frac{1}{2^{n_k}} < D_k < X_t + \frac{1}{2^{n_k}}\right] \leq \frac{2n_k}{2^{n_k}},$$

because $n_k \leq p_k$. Now note that for all $k \geq 1$ and $0 \leq i \leq d_k$ we have $X_{n_k+i} \in \mathcal{D}_{p_k}$, while almost surely, all other random variables do not fall in $\mathcal{D}$. Then, denote by $\mathcal{E}$ the event of probability 1 where $\mathbb{X}$ does not visit $\mathcal{D}$ except for times $n_k + i$ for $k \geq 1$ and $0 \leq i \leq d_k$. In other words,

$$\mathcal{E} := \{X_{n_k+i} \notin \mathcal{D}, \quad k \geq 1, d_k < i < n_{k+1} - n_k\}$$

and $\mathbb{P}(\mathcal{E}) = 1$. We also denote by $\mathcal{A}_k$ the event $\mathcal{A}_k := \{\min_{t<n_k} |X_t - D_k| \geq 2^{-n_k}\}$ and $\mathcal{B}_k$ the event $\mathcal{B}_k := \{|U_k - D_k| \geq 2^{-k}\}$. We have $\mathbb{P}(\mathcal{B}_k^c) \leq 2^{-k+1}$ and we showed previously $\mathbb{P}(\mathcal{A}_k^c) \leq \frac{2n_k}{2^{n_k}}$. Now note that $\frac{d_k}{2^{p_k-1}} = o(\frac{1}{2^{n_k+2n_{k+1}+k+1}})$. Therefore, let k_0 such that for any $k \geq k_0$, $\frac{d_k}{2^{p_k-1}} \leq \frac{1}{2^{n_k+2n_{k+1}+k+1}}$. Then, for any $k \geq k_0$, on the event $\mathcal{A}_k \cap \mathcal{B}_k \cap \mathcal{E}$, for any $1 \leq j < n_{k+1} - n_k - d_k$, the $d_k + 1$ nearest neighbors of $X_{n_k+d_k+j}$ are exactly the points $\{X_{n_k+i} = D_{k,i+1},\ 0 \leq i \leq d_k\}$. Indeed,

$$|X_{n_k+d_k+j} - D_{k,i}| \leq |X_{n_k+d_k+j} - D_k| + \frac{d_k}{2^{p_k-1}} \leq \frac{1}{2^{n_k} 4^j} + \frac{1}{2^{n_k+2j}} < \frac{1}{2^{n_k+2j-1}}.$$

Further, for all $t < n_k$,

$$|X_{n_k+d_k+j} - X_t| \geq |D_k - X_t| - |X_{n_k+d_k+j} - D_k| \geq \frac{1}{2^{n_k}} - \frac{1}{2^{n_k+2}} > \frac{1}{2^{n_k+2j-1}}.$$

and finally, for $1 \leq j' < j$ and any $0 \leq i \leq d_k$, we have

$$\begin{aligned}
|X_{n_k+d_k+j} - X_{n_k+d_k+j'}| &\geq |X_{n_k+d_k+j} - X_{n_k+d_k+j-1}| = 3 \cdot \frac{|U_k - D_k|}{2^{n_k+2j}} \\
&\geq |X_{n_k+d_k+j} - D_k| + 2 \cdot \frac{1}{2^{n_k+2j+k}} \\
&\geq |X_{n_k+d_k+j} - D_k| + 2 \cdot \frac{d_k}{2^{p_k-1}} \\
&> |X_{n_k+d_k+j} - D_k| + |D_k - D_{k,i}| \\
&\geq |X_{n_k+d_k+j} - D_{k,i}|.
\end{aligned}$$

We now observe that

$$\max_{n_k+d_k+1\leq t<n_{k+1}} k_t = o\left(\frac{n_{k+1}}{(\log n_k)^{1+\delta}}\right) = o(d_k).$$

Therefore, let k_1 such that for any $k \geq k_1$, and any $1 \leq j < n_{k+1} - n_k - d_k$, we have $k_{n_k+d_k+j} \leq d_k$. Now for any $k \geq \max(k_0, k_1)$, on the event $\mathcal{A}_k \cap \mathcal{B}_k \cap \mathcal{E}$, $(k_t)_t$NN makes an error in the prediction of all $X_{n_k+d_k+j}$ for $1 \leq j < n_{k+1}-n_k-d_k$ since its k_t closest neighbors are in the set $\{X_{n_k+i} = D_{k,i+1},\ 0 \leq i \leq d_k\}$ which all have value $\mathbb{1}_{\mathcal{D}}(X_{n_k+i}) = 1$ instead of $\mathbb{1}_{\mathcal{D}}(X_{n_k+d_k+j}) = 0$.

Last, note that the frequency of the times of the form $n_k + i$ for $k \geq 1$ and $0 \leq i \leq d_k$ vanishes to 0, because $d_k = o(n_{k+1} - n_k)$ and $n_{k+1} \sim n_k$. Therefore, on the event $\mathcal{E} \cap \cup_{k'\geq 1} \bigcap_{k\geq k'} (\mathcal{A}_k \cap \mathcal{B}_k)$, the learning rule $(k_t)_t$NN has error rate $\mathcal{L}_{\mathbb{X}}((k_t)_t NN, f^*) = 1$. Now note that $\mathbb{P}[\mathcal{E} \cap \mathcal{A}_k^c \cap \mathcal{B}_k^c] \leq 2^{-k+1} + \frac{2n_k}{2^{n_k}}$. Because we have $\sum_{n\geq 1} 2^{-n+1} < \infty$ and $\sum_{n\geq 1} \frac{2n}{2^n} < \infty$, the Borel-Cantelli lemma implies $\mathbb{P}[\mathcal{E} \cap \cup_{k'\geq 1} \bigcap_{k\geq k'} (\mathcal{A}_k \cap \mathcal{B}_k)] = 1$. To summarize, with probability one, $(k_t)_t$NN has error rate 1 hence is not consistent for process $\mathbb{X}$ and target function $f^* = \mathbb{1}_{\mathcal{D}}$. This ends the proof that $(k_t)_t$NN is not universally consistent for process $\mathbb{X}$.

We now show that $\mathbb{X} \in$ Soul by showing that in fact $\mathbb{X} \in$ CRF. Let $A \subset [0, 1]$. We will show that the frequencies of falling in A converge almost surely to $\mu(A)$ where μ is the Lebesgue measure. We introduce the random variables

$$Y_k = \sum_{i=0}^{n_{k+1}-n_k-1} \mathbb{1}_A(X_{n_k+i}).$$

Again, for $k \geq 1$, and $1 \leq j < n_{k+1} - n_k - d_k$, $X_{n_k+d_k+j}$ is an absolutely continuous random variable with density $f(x) = \frac{1}{2^{p_k-1}} \sum_{l=0}^{2^{p_k-1}-1} f_l(x)$ where $f_l(x)$ corresponds to the conditional density to $D_k = \frac{2l+1}{2^{p_k}} =: d_l$, i.e.

$$f_l(x) = 2^{n_k} 4^j \cdot \mathbb{1}\left(x \in \left[d_l - \frac{d_l}{2^{n_k}4^j}, d_l + \frac{1-d_l}{2^{n_k}4^j}\right]\right)$$

But $x \in \left[d_l - \frac{d_l}{2^{n_k}4^i}, d_l + \frac{1-d_l}{2^{n_k}4^i}\right]$ i.if $\frac{2^{p_k-1}(x-\frac{1}{2^{n_k}4^i})}{1-\frac{1}{2^{n_k}4^i}} - \frac{1}{2} \leq l \leq \frac{2^{p_k-1}x}{1-\frac{1}{2^{n_k}4^i}} - \frac{1}{2}$. Therefore, the number $N(x)$ of non-zero terms in the sum $f(x) = \frac{1}{2^{p_k-1}} \sum_{l=0}^{2^{p_k-1}-1} f_l(x)$ is

$$\frac{2^{p_k-1-n_k-2i}}{1-\frac{1}{2^{n_k}4^i}} - 1 \leq N(x) \leq \frac{2^{p_k-1-n_k-2i}}{1-\frac{1}{2^{n_k}4^i}} + 1$$

Hence,

$$\left|f(x) - \frac{1}{1-\frac{1}{2^{n_k}4^i}}\right| = \left|\frac{2^{n_k}4^i N(x)}{2^{p_k-1}} - \frac{1}{1-\frac{1}{2^{n_k}4^i}}\right| \leq \frac{1}{2^{p_k-1-n_k-2i}}.$$

Finally, we obtain

$$|\mathbb{P}(X_{n_k+d_k+j} \in A) - \mu(A)| \leq \frac{1}{2^{p_k-1-n_k-2j}} + \frac{1}{2^{n_k+2j}-1}.$$

Therefore,

$$\begin{aligned}|\mathbb{E}Y_k - (n_{k+1}-n_k)\mu(A)| &\leq \sum_{i=0}^{d_k} |\mathbb{P}(X_{n_k+i}\in A) - \mu(A)| + \sum_{j=1}^{n_{k+1}-n_k-d_k-1} \mathbb{P}(X_{n_k+d_k+j}\in A)\\ &\leq d_k + 1 + \frac{n_{k+1}-n_k}{2^{p_k-2n_{k+1}}} + \frac{n_{k+1}-n_k}{2^{n_k}-1}\\ &\leq d_k + C\end{aligned}$$

where $C \geq 1$ is some universal constant, given that $\frac{n_{k+1}-n_k}{2^{p_k-2n_{k+1}}} \to 0$ and $\frac{n_{k+1}-n_k}{2^{n_k}-1} \to 0$ as $k \to \infty$. Now note that because Y_k is a sum of $n_{k+1}-n_k$ random variables bounded by 1, then

$$Var(Y_k) \leq (n_{k+1}-n_k)^2 = \mathcal{O}\left(\frac{n_{k+1}^2}{k^{1+2\epsilon}}\right).$$

Therefore, $\sum_{k\geq 1} \frac{Var(Y_k)}{(n_{k+1}-1)^2} < \infty$. Further, we can note that the random variables $(Y_k)_{k\geq 1}$ are together independent. Thus, by Kolmogorov's Convergence Criteria, we obtain

$$\sum_{l=1}^{k} \frac{Y_l - \mathbb{E}Y_l}{n_{k+1}-1} \to 0 \quad (a.s.)$$

We then apply Kronecker's lemma which gives

$$\epsilon_k := \frac{\sum_{l=1}^{k} Y_l - \mathbb{E}Y_l}{n_{k+1}-1} \to 0 \quad (a.s.)$$

We now compute,

$$\begin{aligned}\left|\frac{1}{n_{k+1}-1}\sum_{t=1}^{n_{k+1}-1} \mathbb{1}_A(X_t) - \mu(A)\right| &= \frac{1}{n_{k+1}-1}\left|\sum_{l=1}^{k} Y_l - (n_{k+1}-n_k)\mu(A)\right|\\ &= \frac{1}{n_{k+1}-1}\left|(n_{k+1}-1)\epsilon_k + \sum_{l=1}^{k}\mathbb{E}Y_l - (n_{k+1}-n_k)\mu(A)\right|\\ &\leq \epsilon_k + \frac{Ck + \sum_{l=1}^{k} d_l}{n_{k+1}-1}.\end{aligned}$$

Because $\frac{k}{n_{k+1}-1} \to 0$ and $\sum_{l=1}^{k} d_l = o(n_{k+1}-1)$, we obtain $\frac{1}{n_{k+1}-1}\sum_{t=1}^{n_{k+1}-1}\mathbb{1}_A(X_t) \to \mu(A) \quad (a.s.)$. We complete the proof by noting that for any $n_k \leq T < n_{k+1}$,

$$\frac{1}{n_{k+1}-1}\sum_{t=1}^{n_k-1}\mathbb{1}_A(X_t) \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}_A(X_t) \leq \frac{1}{n_k-1}\sum_{t=1}^{n_{k+1}-1}\mathbb{1}_A(X_t),$$

and that $\frac{n_k-1}{n_{k+1}-1} \to 1$ as $k \to \infty$. Therefore $\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}_A(X_t) \to \mu(A) \quad (a.s.)$. This shows that $\mathbb{X} \in \text{CRF}$. Because $\text{CRF} \subset \text{SOUL}$ [Han21a], this ends the proof of the theorem. ■

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T
Output: Predictions $\hat{Y}_t = kC1NN_t(\boldsymbol{X}_{<t}, \boldsymbol{Y}_{<t}, X_t)$ for $t \leq T$
$\hat{Y}_1 := 0$
$\mathcal{D}_2 := \{1\}$
$n_1 \leftarrow 0$
$t \leftarrow 2$
while $t \leq T$ **do**
  if *exists* $u < t$ *such that* $X_u = X_t$ **then**
    $\hat{Y}_t := Y_u$
    $\mathcal{D}_{t+1} := \mathcal{D}_t$
  else
    $\phi(t) := \arg\min_{u \in \mathcal{D}_t} \rho(X_t, X_u)$
    $\hat{Y}_t := Y_{\phi(t)}$
    $n_{\phi(t)} \leftarrow n_{\phi(t)} + 1$
    $n_t \leftarrow 0$
    if $n_{\phi(t)} = k$ **then**
      $\mathcal{D}_{t+1} := (\mathcal{D}_t \setminus \{\phi(t)\}) \cup \{t\}$
    else
      $\mathcal{D}_{t+1} := \mathcal{D}_t \cup \{t\}$
    end
  end
  $t \leftarrow t + 1$
end

Algorithm 3.1: kC1NN learning rule

3.5 An Optimistically Universal Learning Rule

In this section, we present an optimistically universal algorithm and give a characterization of SOUL. We start by defining our new learning rule $k-$Capped $1-$Nearest Neighbor (kC1NN) for any $k \geq 2$. This is a simple variant of the traditional 1NN learning rule where kC1NN performs the 1NN learning rule over a reduced training set. Recall that in the 1NN learning rule, we assign to the new input X_t the value of the nearest neighbor $Y_{NN(t)}$ where $NN(t) = \arg\min_{u<t} \rho(X_t, X_u)$. We refer to the input point $X_{NN(t)}$ as the representant of the input value X_t. In the kC1NN learning rule, we keep in memory the number of times n_t each point X_t is used as a representant for following input data and cap this value at k. Precisely, at each step t we update the dataset $\mathcal{D}_t \subset \{u,\ u < t\}$ containing the indices of data points on which 1NN may be performed. To do so, when n_u reaches k for some $u < t$, we delete u from the current dataset $\mathcal{D}_t$. At each iteration, if the input X_t has already been visited, we use simple memorization to predict Y_t, we do not update the values $(n_u)_{u<t}$ and do not include t in the dataset $\mathcal{D}_{t+1}$. Otherwise, kC1NN performs the 1NN learning rule on the current dataset $(X_u, Y_u)_{u \in \mathcal{D}_t}$, where ties can be broken arbitrarily for instance with minimum index, and updates $(n_u)_{u \in \mathcal{D}_t}$ and the dataset accordingly. In the following, we denote by $\phi(t)$ the index of the representant used for X_t, i.e. of its closest neighbor within the dataset $\mathcal{D}_t$. The rule is formally described in Algorithm 3.1.

In Section 3.4 we presented a process $\mathbb{X}$ on which nearest neighbor fails. The main reason for this failure is that some specific input points X_t can be used an arbitrarily large number of times as representant for future points, thereby inducing a large number of prediction errors. The learning rule kC1NN is designed precisely to tackle this issue by ensuring that any datapoint X_t for $t \geq 1$ is used at most k times as representant, i.e., $|\{u > t : \ \phi(u) = t\}| \leq k$. The goal of this section is to show that 2C1NN is optimistically universal for general separable Borel instance space $(\mathcal{X}, \rho)$ and near-metric separable value space $(\mathcal{Y}, \ell)$ with bounded loss. To provide a simpler exposition of the result, we now show that kC1NN is in fact optimistically universal for $k \geq 4$ starting with $\mathcal{X} = [0, 1]$. This will in turn give the result for general standard Borel space as shown in Section 3.5.2 and already provides all the intuitions necessary for the general case presented in Section 3.6.

3.5.1 Universal online learning for contexts in $[0, 1]$

We consider the case $\mathcal{X} = [0, 1]$ and for binary classification in this section and show that 4C1NN is optimistically universal for this input space. To do so, we prove that 4C1NN is universally consistent under all processes in $\text{SMV}_{([0,1],|\cdot|)}$ which yields $\text{SMV}_{([0,1],|\cdot|)} \subset \text{SOUL}_{([0,1],|\cdot|),(\{0,1\},\ell_{01})}$. Together with Proposition 3.1, this will show $\text{SOUL}_{([0,1],|\cdot|),(\{0,1\},\ell_{01})} = \text{SMV}_{([0,1],|\cdot|)}$ and as a result, that 4C1NN is optimistically universal. As a first step, we focus on the simple function f^* represented by the fixed interval $[0, 1/2]$ in the binary classification setting, and show that 4C1NN is consistent under any input process for this target function.

Proposition 3.2. *Let $\mathcal{X} = [0, 1]$ with the usual topology. We consider the binary classification setting $\mathcal{Y} = \{0, 1\}$ with ℓ_{01} binary loss. Under any input process $\mathbb{X} \in SMV_{([0,1],|\cdot|)}$, the learning rule 4C1NN is strongly consistent for the target function $f^* = \mathbb{1}_{[0,1/2]}$.*

Proof We reason by the contrapositive and suppose that 4C1NN is not consistent on f^*. We will show that the process $\mathbb{X}$ disproves the $\text{SMV}_{([0,1],|\cdot|)}$ condition by considering the partition $\mathcal{P}$ of $\mathcal{X}$ defined by

$$\left\{\frac{1}{2}\right\} \cup \bigcup_{k\geq 1} \left[\frac{1}{2} - \frac{1}{2k}; \frac{1}{2} - \frac{1}{2(k+1)}\right) \cup \bigcup_{k\geq 1} \left(\frac{1}{2} + \frac{1}{2(k+1)}; \frac{1}{2} + \frac{1}{2k}\right].$$

Precisely, we will show that the process does not visit a sublinear number of sets of this partition with nonzero probability.

Because 4C1NN is not consistent, $\delta := \mathbb{P}(\mathcal{L}_{\mathbb{X}}(4C1NN, f^*) > 0) > 0$. Define

$$\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}(4C1NN, f^*) > 0\}.$$

We now consider a specific realization $\boldsymbol{x} = (x_t)_{t\geq 0}$ of the process $\mathbb{X}$ falling in the event $\mathcal{A}$. Note that $\boldsymbol{x}$ is not random anymore. We now show that $\boldsymbol{x}$ does not visit a sublinear number of sets in the partition $\mathcal{P}$. By construction $\epsilon := \mathcal{L}_{\boldsymbol{x}}(4C1NN, f^*) > 0$. We now denote by $(t_k)_{k\geq 1}$ the increasing sequence of all times when 4C1NN makes an error in the prediction of $f^*(x_t)$. Now define an increasing sequence of times $(T_l)_{l\geq 1}$ such that

$$\frac{1}{T_l} \sum_{t=1}^{T_l} \ell_{01}(4C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) > \frac{\epsilon}{2}.$$

For any $l \geq 1$ consider the last index $k = \max\{u, t_u \leq T_l\}$ when 4C1NN makes a mistake. Then we obtain $k > \frac{\epsilon}{2} T_l \geq \frac{\epsilon}{2} t_k$. Considering the fact that $(T_l)_{l \geq 1}$ is an increasing unbounded sequence we therefore obtain an increasing sequence of indices $(k_l)_{l \geq 1}$ such that $t_{k_l} < \frac{2k_l}{\epsilon}$.

At an iteration where the new input x_t has not been previously visited we will denote by $\phi(t)$ the index of the nearest neighbor of the current dataset in the 4C1NN learning rule. Now let $l \geq 1$. We focus on the time t_{k_l}. Consider the tree $\mathcal{G}$ where nodes are times $\mathcal{T} := \{t,\ t \leq t_{k_l},\ x_t \notin \{x_u, u < t\}\}$ for which a new input was visited, where the parent relations are given by $(t, \phi(t))$ for $t \in \mathcal{T} \setminus \{1\}$. In other words, we construct the tree in which a new input is linked to its representant which was used to derive the target prediction. Note that by definition of the 4C1NN learning rule, each node has at most 4 children and a node is not in the dataset at time t_{k_l} when it has exactly 4 children.

By symmetry, we will suppose without loss of generality that the majority of input points on which 4C1NN made a mistake belong to the first half $[0, \frac{1}{2}]$ i.e.

$$|\{t \leq t_{k_l},\ \ell_{01}(4C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) = 1,\ x_t \in [0, 1/2]\}| \geq \frac{k_l}{2}$$

or equivalently, $\left|\left\{k \leq k_l,\ x_{t_k} \leq \frac{1}{2}\right\}\right| \geq \frac{k_l}{2}$.

Let us now consider the subgraph $\tilde{\mathcal{G}}$ given by restricting $\mathcal{G}$ only to nodes in the first half-space $[0, 1/2]$ which are mapped to the true value 1 i.e. on times $\{t \in \mathcal{T},\ x_t \leq \frac{1}{2}\}$. In this subgraph, the only times with no parent are times t_k with $k \leq k_l$ and $x_{t_k} \leq \frac{1}{2}$ and possibly time $t = 1$. Indeed, if a time in $\tilde{\mathcal{G}}$ has a parent $\phi(t)$ in $\tilde{\mathcal{G}}$, the prediction of 4C1NN for x_t returned the correct answer 1. The converse is also true except for the root time $t = 1$ which has no parent in $\mathcal{G}$. Therefore, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with roots times $\{t_k,\ k \leq k_l,\ x_{t_k} \leq \frac{1}{2}\}$ (and possibly $t = 1$). For a given time t_k with $k \leq k_l$ and $x_{t_k} \leq \frac{1}{2}$, we will denote by $\mathcal{T}_k$ the corresponding tree in $\tilde{\mathcal{G}}$ with root t_k. We say that the $\mathcal{T}_k$ is a *good* tree if all times $t \in \mathcal{T}_k$ of this tree are parent in $\mathcal{G}$ to at most 1 time from the second half-space $(\frac{1}{2}, 1]$ i.e. if

$$\forall t \in \mathcal{T}_k, \quad \left|\left\{u \leq t_{k_l},\ \phi(u) = t,\ x_u > \frac{1}{2}\right\}\right| \leq 1.$$

We denote by $G = \{k \leq k_l,\ x_{t_k} \leq \frac{1}{2},\ \mathcal{T}_k \text{ good}\}$ the set of indices of good trees. By opposition, we will say that a tree is *bad* otherwise. We now give a simple upper bound on N_{bad} the number of bad trees. Note that for any time $t \in \mathcal{T}_k$ of a tree, times in $\left\{u \leq t_{k_l},\ \phi(u) = t,\ x_u > \frac{1}{2}\right\}$ are when 4C1NN makes a mistake on the second-half $(\frac{1}{2}, 1]$. Therefore,

$$\sum_{k \leq k_l,\, x_{t_k} \leq \frac{1}{2}} \sum_{t \in \mathcal{T}_k} \left|\left\{u < t_{k_l},\ \phi(u) = t,\ x_u > \frac{1}{2}\right\}\right| \leq \left|\left\{k \leq k_l, x_{t_k} > \frac{1}{2}\right\}\right| \leq \frac{k_l}{2}$$

because by hypothesis $\left|\left\{k \leq k_l,\ x_{t_k} \leq \frac{1}{2}\right\}\right| \geq \frac{k_l}{2}$. Therefore, since each bad tree contains a node which is parent to at least 2 times of mistake in $(\frac{1}{2}, 1]$, we obtain

$$N_{\text{bad}} \leq \sum_{k \leq k_l,\, x_{t_k} \leq \frac{1}{2}} \sum_{t \in \mathcal{T}_k} \frac{1}{2} \left|\left\{u < t_{k_l},\ \phi(u) = t,\ x_u > \frac{1}{2}\right\}\right| \leq \frac{k_l}{4}.$$

Thus, the number of good trees is $|G| = \left|\left\{k \leq k_l,\ x_{t_k} \leq \frac{1}{2}\right\}\right| - N_{\text{bad}} \geq \frac{k_l}{4}$. We now focus on good trees only and analyze their relation with the final dataset $\mathcal{D}_{t_{k_l}}$. Precisely, for a good tree $\mathcal{T}_k$, denote $\mathcal{V}_k = \mathcal{T}_k \cap \mathcal{D}_{t_{k_l}}$ the set of times which are present in the final dataset and belong to the tree induced by error time t_k. One can note that the sets $\{x_u,\ u \in \mathcal{V}_k\}_{k \in G}$ are totally ordered:

$$\forall k_1 < k_2 \in G,\ \forall t_1 \in \mathcal{T}_{k_1},\ \forall t_2 \in \mathcal{T}_{k_2}, \quad x_{t_1} < x_{t_2}.$$

This can be shown by observing that at each iteration t of 4C1NN, the following invariant is conserved: the sets $\{x_u,\ u \in \mathcal{T}_k \cap \mathcal{D}_t\}_{k \in \{l \in G,\, t_l \leq t\}}$ are totally ordered. The induction follows from the fact that when a new input point is visited, 4C1NN performs the 1NN learning rule on the current dataset $\mathcal{D}_l$. Therefore, either the sets $\{x_u,\ u \in \mathcal{T}_k \cap \mathcal{D}_t\}_{k \in \{l \in G,\, t_l \leq t\}}$ are conserved, or a new point is added when $t = t_k$ for some $k \leq k_l$ which forms its own tree and is closest to $(\frac{1}{2}, 1]$ than all other sets $\{x_u,\ u \in \mathcal{T}_k \cap \mathcal{D}_t\}_{k \in \{l \in G,\, t_l \leq t\}}$, or a new point is added to an existing tree $\mathcal{T}_k$ in which case it should be closer to some time of $\mathcal{T}_k \cap \mathcal{D}_t$ than any time in $\mathcal{T}_{k-1} \cap \mathcal{D}_t$ or $\mathcal{T}_{k+1} \cap \mathcal{D}_t$—if $\mathcal{T}_{k-1}$ or $\mathcal{T}_{k+1}$ exist. Additionally, a time may be removed which is still consistent with the invariant. Last, we observe that these sets never run empty because a time is removed only when at least 3 other points were added to the same set.

We now reason by induction to show that the sets $\{x_u,\ u \in \mathcal{V}_k\}_{k \in G}$ are also well separated—in a multiplicative way. Let us order the good trees by $G = \{g_1 < \ldots < g_{|G|}\}$ and start with tree $\mathcal{T}_{g_1}$. Consider any leaf of this tree and the corresponding path to the root $p_l \to p_{l-1} \to p_0 = t_{g_1}$ and define $x^1 = \min_{1 \leq i \leq l} x_{p_i}$. By construction, any point on this path is being replaced by its parent. Therefore, at any step of the algorithm 4C1NN at least one point on this path is available in the dataset $\mathcal{D}_t$ for any $t \geq t_{g_1}$—for instance the last time p_i such that $p_i \leq t$. This point x^1 provides a lower bound for the maximum point in $\{x_u,\ u \in \mathcal{T}_{g_1} \cap \mathcal{D}_t\}$ which in turn will provide a lower bound for all points in $\{x_u,\ u \in \mathcal{T}_{g_2} \cap \mathcal{D}_t\}$.

Let us now turn to $\mathcal{T}_{g_2}$. By construction, in a good tree $\mathcal{T}_k$, a time $t \in \mathcal{T}_k$ which is not in the final dataset $\mathcal{D}_{t_{k_l}}$ must be parent to at least 3 other times within $\mathcal{T}_k$. Therefore, until the minimal depth of an available time $\mathcal{V}_{g_2} = \mathcal{T}_{g_2} \cap \mathcal{D}_{t_{k_l}}$ in the current dataset $\mathcal{D}_{t_{k_l}}$, each node of the tree $\mathcal{T}_{g_2}$ has at least 3 parents which correspond necessarily to times $t > t_{g_2}$. Therefore, the minimal depth $d(g_2)$ of an available time $\mathcal{V}_k$ in the current dataset satisfies

$$\sum_{i=0}^{d(g_2)-1} 3^i \leq |\mathcal{T}_{g_2}| \leq t_{k_l}.$$

Therefore $d(g_2) \leq \log_3(2t_{k_l} + 1) \leq \log_3 t_{k_l}$. Now consider the specific path from this node in $\mathcal{V}_{g_2}$ of minimal depth to the root t_{g_2}. Denote this path $p_{d(g_2)} \to p_{d(g_2)-1} \to p_0 = t_{g_2}$. Each arc of this path represents the fact that at the corresponding iteration p_i of 4C1NN, the parent $x_{p_{i-1}}$ was closer from x_{p_i} than any other point of the current dataset $\mathcal{D}_{p_i}$, in particular any point of $\{x_u,\ u \in \mathcal{T}_{g_1} \cap \mathcal{D}_{p_i}\}$. This gives $|x_{p_{i-1}} - x_{p_i}| \leq |x^1 - x_{p_{i-1}}| = x_{p_{i-1}} - x^1$ because we have $x_{p_{i-1}}, x_{p_i} > x^1$. Therefore we obtain

$$x_{p_{i-1}} \geq \frac{x^1 + x_{p_i}}{2}.$$

Indeed, if this were not the case we would have $|x_{p_{i-1}} - x_{p_i}| = x_{p_i} - x_{p_{i-1}} > x_{p_{i-1}} - x^1$. Similarly, considering the fact that 4C1NN makes a mistake at time t_{g_2}, the parent of t_{g_2}

satisfies $x_{\phi(t_{g_2})} > \frac{1}{2}$ which yields $x_{t_{g_2}} \geq \frac{x^1 + x_{\phi(t_{g_2})}}{2} \geq \frac{x^1 + \frac{1}{2}}{2}$. Hence, for any $0 \leq i \leq d(g_2)$,

$$x_{p_i} \geq x^1 \left(1 - \frac{1}{2^i}\right) + \frac{x_{t_{g_2}}}{2^i} \geq x^1 + \frac{x_{t_{g_2}} - x^1}{2^{d(g_2)}} \geq x^1 + \left(\frac{1}{2} - x^1\right) t_{k_l}^{-\frac{\log 2}{\log 3}}.$$

Again, at every iteration $t \geq t_{g_2}$ of 4C1NN, at least one of the points x_{p_i} is available in the dataset $\mathcal{D}_t$—for instance the last x_{p_i} such that $p_i \leq t$. By total ordering, this $x^2 := \min_{0 \leq i \leq d(g_2)} x_{p_i}$ provides a lower bound for all points $\{x_u,\ u \in \mathcal{T}_{g_3} \cap \mathcal{D}_t\}$ whenever $t \geq t_{g_3}$. Hence, the lower bound x^2 acts as a new barrier: the equivalent of x^1 for the above argument with $\mathcal{T}_{g_2}$.

For clarity, we precise the next iteration of the induction for $\mathcal{T}_{g_3}$. The minimal depth $d(g_3)$ of an available time $\mathcal{V}_{g_3}$ satisfies $d(g_3) \leq \log_3(t_{k_l} - t_{g_3} + 1) + 1$ using the same argument as above. Now consider the corresponding path in $\mathcal{T}_{g_3}$ from this minimal depth node to the root $p_{d(g_3)} \to \ldots \to p_0 = t_{g_3}$. By definition of the 4C1NN learning rule, the parent $x_{p_{i-1}}$ was closer to x_{p_i} than any point of $\{x_u,\ u \in \mathcal{T}_{g_2} \cap \mathcal{D}_t\}$. By the previous step of the induction, we know that the maximum value of this set is at least x^2. Therefore, we obtain $|x_{p_{i-1}} - x_{p_i}| \leq |x^2 - x_{p_i}| = x_{p_i} - x^2$. We recall that we also have $x_{p_{i-1}} \geq x^2$ and $x_{p_i} \geq x^2$. The same argument as above gives $x_{p_i} \geq \frac{x^2 + x_{p_{i-1}}}{2}$. Further, we obtain similarly $x_{t_{g_3}} \geq \frac{x^2 + x_{\phi(t_{g_3})}}{2} \geq \frac{x^2 + \frac{1}{2}}{2}$. Hence, for all $0 \leq i \leq d(g_3)$,

$$x_{p_i} \geq x^2 + \frac{x_{t_{g_3}} - x^2}{2^{d(g_3)}} \geq x^2 + \left(\frac{1}{2} - x^2\right) t_{k_l}^{-\frac{\log 2}{\log 3}}.$$

We denote $x^3 := \min_{0 \leq i \leq d(g_3)} x_{p_i}$, which now acts as a lower barrier for the tree $\mathcal{T}_{g_4}$ and we can apply the induction.

We complete this induction for $\mathcal{T}_{g_3}, \ldots, \mathcal{T}_{g_{|G|}}$. This creates a sequence of distinct visited input points $(x^i)_{1 \leq i \leq |G|}$ with $x^i \leq \frac{1}{2}$ such that for any $1 \leq i < |G|$, $x^{i+1} \geq x^i + \left(\frac{1}{2} - x^i\right) t_{k_l}^{-\frac{\log 2}{\log 3}}$ i.e.

$$\frac{1}{2} - x^{i+1} \leq \left(\frac{1}{2} - x^i\right)\left(1 - t_{k_l}^{-\frac{\log 2}{\log 3}}\right).$$

In particular, we can observe that $0 \leq x^1 < x^2 < \ldots < x^{|G|} \leq \frac{1}{2}$. Further, recalling that we have $t_{k_l} < \frac{2k_l}{\epsilon}$, we get

$$\log\left(\frac{1}{2} - x^{i+1}\right) - \log\left(\frac{1}{2} - x^i\right) \leq \log\left(1 - t_{k_l}^{-\frac{\log 2}{\log 3}}\right) \leq -t_{k_l}^{-\frac{\log 2}{\log 3}} \leq -\left(\frac{\epsilon}{2k_l}\right)^{\frac{\log 2}{\log 3}},$$

for any $1 \leq i \leq |G| - 1$. We will now argue that most of these points x^i fall in distinct sets of the type $[a_k, a_{k+1})$ where $a_k := \frac{1}{2} - \frac{1}{2k}$ for $k \geq 1$. We observe that for any $k \geq 1$, we have by concavity $\log\left(\frac{1}{2} - a_{k+1}\right) - \log\left(\frac{1}{2} - a_k\right) = \log\left(1 - \frac{1}{k+1}\right) \geq -\frac{\log 2}{k+1}$. Therefore, with $k^0 = \left\lceil \log 2 \cdot \left(\frac{2k_l}{\epsilon}\right)^{\frac{\log 2}{\log 3}} \right\rceil$, for any $k \geq k^0$ we have

$$\log\left(\frac{1}{2} - a_{k+1}\right) - \log\left(\frac{1}{2} - a_k\right) > -\left(\frac{\epsilon}{2k_l}\right)^{\frac{\log 2}{\log 3}}.$$

Therefore, for any $1 \leq i \leq |G| - 1$ such that $x^i > a_{k^0}$, x^i and x^{i+1} would lie in different sets of the type $[a_k, a_{k+1})$, $k \geq 1$. In fact because the sequence $(x^i)_{1 \leq i \leq |G|}$ is increasing, if $x^{i^*} > a_{k^0}$ then all points $(x^i)_{i^* \leq i \leq |G|}$ lie in distinct sets of the type $[a_k, a_{k+1})$, $k \geq 1$. Recall that $|G| \geq \frac{k_l}{4}$. Denote $i^* = \lfloor \frac{k_l}{8} \rfloor$. Because $(k_l)_{l \geq 1}$ is an increasing sequence, we have

$$\log\left(\frac{1}{2} - x^{i^*}\right) \leq \log\left(\frac{1}{2}\right) - (i^* - 1)\left(\frac{\epsilon}{2k_l}\right)^{\frac{\log 2}{\log 3}} \underset{l \to \infty}{\sim} -c_\epsilon k_l^{1 - \frac{\log 2}{\log 3}},$$

where $c_\epsilon := \frac{1}{8}\left(\frac{\epsilon}{2}\right)^{\frac{\log 2}{\log 3}}$ is a constant. Therefore,

$$\log\left(\frac{1}{2} - a_{k^0}\right) = -\log(2k^0) \underset{l \to \infty}{\sim} -\frac{\log 2}{\log 3} \log k_l = o\left(\log\left(\frac{1}{2} - x^{i^*}\right)\right)$$

which shows that for some constant l^0 and any $l \geq l^0$ we have $a_{k^0} < x^{i^*} < \frac{1}{2}$. Hence, for any $l \geq l^0$, all the points $(x^i)_{i^* \leq i \leq |G|}$ lie in distinct sets of the partition and there are at least $|G| - \frac{k_l}{8} \geq \frac{k_l}{8}$ such points. Therefore, for any $l \geq l^0$,

$$|\{P \in \mathcal{P}, \quad P \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq \frac{k_l}{8} \geq \frac{\epsilon}{16} t_{k_l}.$$

Because $t_{k_l} \to \infty$ as $l \to \infty$, this shows that $|\{P \in \mathcal{P}, \quad P \cap \boldsymbol{x}_{<T} \neq \emptyset\}| \neq o(T)$. Because this holds for any realization of the event $\mathcal{A}$, we obtained

$$\mathbb{P}(|\{P \in \mathcal{P}, \quad P \cap \mathbb{X}_{<T} \neq \emptyset\}| = o(T)) \leq \mathbb{P}(\mathcal{A}^c) = 1 - \delta < 1.$$

This shows that $\mathbb{X} \notin \mathrm{SMV}_{([0,1],|\cdot|)}$ and ends the proof of the proposition. ■ Note that using the same proof, we observe that the result from Proposition 3.2 holds for all learning rules kC1NN with $k \geq 4$.

We are now ready to prove that 4C1NN is universally consistent under processes of $\mathrm{SMV}_{([0,1],|\cdot|)}$ for the binary classification setting. Intuitively, we analyze the set of functions on which 4C1NN is consistent under a fixed process $\mathbb{X} \in \mathrm{SMV}_{([0,1],|\cdot|)}$ and show that this is a σ-algebra. Proposition 3.2 will be useful to show that this σ-algebra contains all intervals and as a result is the complete Borel σ-algebra $\mathcal{B}$ i.e. 4C1NN is universally consistent under $\mathbb{X}$.

Theorem 3.5. *Let $\mathcal{X} = [0,1]$ with the usual topology $\mathcal{B}$. For the binary classification setting, the learning rule 4C1NN is universally consistent for all processes $\mathbb{X} \in SMV_{([0,1],|\cdot|)}$.*

Proof let $\mathbb{X} \in \mathrm{SMV}_{([0,1],|\cdot|)}$. We will show that 4C1NN is universally consistent on $\mathbb{X}$ by considering the set $\mathcal{S}_\mathbb{X}$ of functions for which it is consistent. More precisely, since $\mathcal{Y} = \{0,1\}$ in the binary setting, all target functions can be described as $f^* = \mathbb{1}_{A_{f^*}}$ where $A_{f^*} = f^{<-1>}(\{1\})$ is a measurable set. In the following, we will refer interchangeably to the function f^* or the set A_{f^*}, and define $\mathcal{S}_\mathbb{X}$ using the corresponding sets:

$$\mathcal{S}_\mathbb{X} := \{A \in \mathcal{B}, \quad \mathcal{L}_\mathbb{X}(4C1NN, \mathbb{1}_A) = 0 \quad (a.s.)\}$$

By construction we have $\mathcal{S}_\mathbb{X} \subset \mathcal{B}$. The goal is to show that in fact $\mathcal{S}_\mathbb{X} = \mathcal{B}$. To do so, we will show that $\mathcal{S}$ satisfies the following properties

- $\emptyset \in \mathcal{S}_{\mathbb{X}}$ and $\mathcal{S}_{\mathbb{X}}$ contains all intervals $[0, s)$ with $0 < s \leq 1$,
- if $A \in \mathcal{S}_{\mathbb{X}}$ then $A^c \in \mathcal{S}_{\mathbb{X}}$ (stable to complementary),
- if $(A_i)_{i\geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$, then $\bigcup_{i\geq 1} A_i \in \mathcal{S}_{\mathbb{X}}$ (stable to $\sigma-$additivity for disjoint sets),
- if $A, B \in \mathcal{S}_{\mathbb{X}}$, then $A \cup B \in \mathcal{S}_{\mathbb{X}}$ (stable to union).

Together, these properties show that $\mathcal{S}_{\mathbb{X}}$ is a $\sigma-$algebra that contains all open intervals of $\mathcal{X} = [0, 1]$. Recall that by definition, $\mathcal{B}$ is the smallest $\sigma-$algebra containing open intervals. Therefore we get $\mathcal{B} \subset \mathcal{S}_{\mathbb{X}}$ which proves the theorem. We now show the four properties.

We start by showing the invariance to complementary. Note that 4C1NN is invariant to labels and that the loss ℓ_{01} is symmetric. Therefore, if it achieves consistency for $\mathbb{1}_A$ it also achieves consistency for $\mathbb{1}_{A^c}$. Indeed, at each step, 4C1NN will use the same representant for the prediction hence for any $t \geq 0$,

$$\ell_{01}(4C1NN(\boldsymbol{x}_{<t}, \mathbb{1}_{\boldsymbol{x}_{<t}\in A}, x_t), \mathbb{1}_{x_t\in A}) = \ell_{01}(4C1NN(\boldsymbol{x}_{<t}, \mathbb{1}_{\boldsymbol{x}_{<t}\in A^c}, x_t), \mathbb{1}_{x_t\in A^c}).$$

4C1NN is clearly consistent for $f^* = 0$. Therefore $\emptyset \in \mathcal{S}_{\mathbb{X}}$. Now let $0 < s \leq 1$. We will show that $[0, s) \in \mathcal{S}_{\mathbb{X}}$. Proposition 3.2 shows that $[0, \frac{1}{2}] \in \mathcal{S}_{\mathbb{X}}$. In fact, one can note that the same proof shows that $[0, \frac{1}{2}) \in \mathcal{S}_{\mathbb{X}}$. Further, for any $0 < s \leq 1$ using the same proof with the following partition centered in s,

$$\{s\} \cup \bigcup_{k\geq 1} \left[s\left(1 - \frac{1}{k}\right); s\left(1 - \frac{1}{k+1}\right)\right) \cup \bigcup_{k\geq 1} \left(s + \frac{1-s}{k+1}; s + \frac{1-s}{k}\right]$$

shows that $[0, s], [0, s) \in \mathcal{S}_{\mathbb{X}}$.

We now turn to the $\sigma-$additivity for disjoint sets. Let $(A_i)_{i\geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$. We denote $A := \bigcup_{i\geq 1} A_i$. We consider the target function $f^* = \mathbb{1}_A$. There are two types of statistical errors: errors of type 1 correspond to $X_t \in A$ and a predicted value 0 while type 2 errors correspond to $X_t \notin A$ and a predicted value 1. We then write the average loss in the following way,

$$\frac{1}{T}\sum_{t=1}^{T} \ell_{01}(4C1NN(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t), f^*(X_t)) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{X_t\in A}\mathbb{1}_{X_{\phi(t)}\notin A} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A},$$

where the first term corresponds to type 1 errors and the second term corresponds to type 2 errors.

We suppose by contradiction that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}(4C1NN, f^*) > 0) := \delta > 0$ Therefore, there exists $\epsilon > 0$ such that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}(4C1NN, f^*) > \epsilon) \geq \frac{\delta}{2}$. We denote this event by $\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}(4C1NN, f^*) > \epsilon\}$. We first analyze the errors induced by one set A_i only. We have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A} + \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i}) &\leq \frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A_i} + \mathbb{1}_{X_t\notin A_i}\mathbb{1}_{X_{\phi(t)}\in A_i}) \\ &= \frac{1}{T}\sum_{t=1}^{T} \ell_{01}(4C1NN(\mathbb{X}_{<t}, \mathbb{1}_{\mathbb{X}_{<t}\in A_i}, X_t), \mathbb{1}_{X_t\in A_i}).\end{aligned}$$

Then, because 4C1NN is consistent for $\mathbb{1}_{A_i}$, we have

$$\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})\to 0 \quad (a.s.).$$

We take $\epsilon_i=\frac{\epsilon}{4\cdot 2^i}$ and $\delta_i=\frac{\delta}{8\cdot 2^i}$. The above equation gives

$$\mathbb{P}\left[\bigcup_{t_0\geq 1}\bigcap_{T\geq t_0}\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})<\epsilon_i\right\}\right]=1.$$

Therefore, let T^i such that

$$\mathbb{P}\left[\bigcap_{T\geq T^i}\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})<\epsilon_i\right\}\right]\geq 1-\delta_i.$$

We will denote by $\mathcal{E}_i$ this event. We now consider the scale of the process $\mathbb{X}_{\leq T^i}$ when falling in A_i, by introducing $\eta_i>0$ such that

$$\mathbb{P}\left[\min_{\substack{t_1,t_2\leq T^i;\ X_{t_1},X_{t_2}\in A_i;\\ X_{t_1}\neq X_{t_2}}}|X_{t_1}-X_{t_2}|>\eta_i\right]\geq 1-\delta_i.$$

We denote by $\mathcal{F}_i$ this event. By the union bound, we have $\mathbb{P}(\bigcup_{i\geq 1}\mathcal{E}_i^c\cup\bigcup_{i\geq 1}\mathcal{F}_i^c)\leq\frac{\delta}{4}$. Therefore, we obtain $\mathbb{P}(\mathcal{A}\cap\bigcap_{i\geq 1}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i)\geq\mathbb{P}(\mathcal{A})-\mathbb{P}(\bigcup_{i\geq 1}\mathcal{E}_i^c\cup\bigcup_{i\geq 1}\mathcal{F}_i^c)\geq\frac{\delta}{4}$. We now construct a partition $\mathcal{P}$ obtained by subdividing each set A_i according to scale η_i. For simplicity, we use the notation $N_i=\lfloor\frac{1}{\eta_i}\rfloor$ and construct the partition given of $\mathcal{X}=[0,1]$ given by

$$\mathcal{P}\ :\quad A^c\cup\bigcup_{i\geq 1}\left\{([N_i\eta_i,1]\cap A_i)\cup\bigcup_{j=0}^{N_i-1}([j\eta_i,(j+1)\eta_i)\cap A_i)\right\}.$$

Let us now consider a realization of $\boldsymbol{x}$ of $\mathbb{X}$ in the event $\mathcal{A}\cap\bigcap_{i\geq 1}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i$. The sequence $\boldsymbol{x}$ is now not random anymore. Our goal is to show that $\boldsymbol{x}$ does not visit a sublinear number of sets in the partition $\mathcal{P}$.

By construction, the event $\mathcal{A}$ is satisfied, therefore there exists an increasing sequence of times $(t_k)_{k\geq 1}$ such that for any $k\geq 1$, $\frac{1}{t_k}\sum_{t=1}^{t_k}\ell_{01}(4C1NN(\boldsymbol{x}_{<t},\mathbb{1}_{\boldsymbol{x}_{<t}\in A},x_t),\mathbb{1}_{x_t\in A})>\frac{\epsilon}{2}$. Therefore, we obtain for any $k\geq 1$,

$$\sum_{i\geq 1}\frac{1}{t_k}\sum_{t=1}^{t_k}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A}+\mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i})>\frac{\epsilon}{2}.$$

Also, because the events $\mathcal{E}_i$ are met, we have

$$\sum_{i\geq 1;\ t_k\geq T^i}\frac{1}{t_k}\sum_{t=1}^{t_k}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A}+\mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i})<\sum_{i\geq 1,t_k\geq T^i}\epsilon_i\leq\frac{\epsilon}{4}.$$

Combining the two above equations gives

$$\frac{1}{t_k}\sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}\left(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A}+\mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}\right)>\frac{\epsilon}{4}. \tag{3.1}$$

We now consider the set of times such that an input point fell into the set A_i with $T^i > t_k$, either creating a mistake in the prediction of 4C1NN or inducing a later mistake within time horizon t_k: $\mathcal{T} := \bigcup_{i\geq 1;\ T^i>t_k} \mathcal{T}_i$ where

$$\mathcal{T}_i := \left\{t \leq t_k,\ x_t \in A_i,\ \left(x_{\phi(t)} \notin A \text{ or } \exists t < u \leq t_k \text{ s.t. } \phi(u) = t,\ x_u \notin A\right)\right\}.$$

We now show that all points x_t for $t \in \mathcal{T}$ fall in distinct sets of the partition $\mathcal{P}$. Indeed, because the sets A_i are disjoint, it suffices to check that for any $i \geq 1$ such that $T^i > t_k$, the points x_t for $t \in \mathcal{T}_i$ fall in distinct of the following sets

$$[N_i\eta_i, 1] \cap A_i, \quad [j\eta_i, (j+1)\eta_i) \cap A_i, \quad 0 \leq j \leq N_i - 1.$$

Note that for any $t_1 < t_2 \in \mathcal{T}_i$ we have $x_{t_1}, x_{t_2} \in A_i$ and $x_{t_1} \neq x_{t_2}$. Indeed, we cannot have $x_{t_2} = x_{t_1}$ otherwise 4C1NN would make no mistake at time t_2 and x_{t_2} would induce no future mistake either (recall that if an input point was already visited, we use simple memorization for the prediction and do not add it to the dataset). Therefore, because the event $\mathcal{F}_i$ is satisfied, for any $t_1 < t_2 \in \mathcal{T}_i$ we have $|x_{t_1} - x_{t_2}| > \eta_i$. Hence x_{t_1} and x_{t_2} lie in different sets among $[N_i\eta_i, 1] \cap A_i$ or $[j\eta_i, (j+1)\eta_i) \cap A_i$ for $0 \leq j \leq N_i - 1$. This shows that all points $\{x_t,\ t \in \mathcal{T}\}$ lie in different sets of the partition $\mathcal{P}$. Therefore,

$$|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}|.$$

We now lower bound $|\mathcal{T}|$, which will uncover the main interest of the learning rule 4C1NN. Intuitively, this learning rule prohibits a single input point x_t to induce a large number of mistakes in the learning process. Indeed, any input point incurs at most $1 + 4 = 5$ mistakes while this number of mistakes incurred by a single point can potentially by unbounded for the traditional 1NN learning rule. We now formalize this intuition.

$$\begin{aligned}
&\sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}\left(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A}+\mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}\right)\\
&\qquad=\sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}\left(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A}+\sum_{t<u\leq t_k}\mathbb{1}_{x_u\notin A}\mathbb{1}_{x_t\in A_i}\mathbb{1}_{\phi(u)=t}\right)\\
&\qquad=\sum_{i\geq 1;\ T^i>t_k}\sum_{t\leq t_k,\ x_t\in A_i}\left(\mathbb{1}_{x_{\phi(t)}\notin A}+\sum_{t<u\leq t_k}\mathbb{1}_{x_u\notin A}\mathbb{1}_{\phi(u)=t}\right)\\
&\qquad\leq\sum_{i\geq 1;\ T^i>t_k}\sum_{t\leq t_k,\ x_t\in A_i}5\max\left(\mathbb{1}_{x_{\phi(t)}\notin A},\ \mathbb{1}_{x_u\notin A}\mathbb{1}_{\phi(u)=t},\ t<u\leq t_k\right)\\
&\qquad=5|\mathcal{T}|
\end{aligned}$$

where in the last inequality we used the fact that a given time t can have at most 4 children i.e. $|\{u > t, \phi(u) = t\}| \leq 4$ with the 4C1NN learning rule. We now use Eq (3.1) to obtain

$$|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}| \geq \frac{\epsilon}{20} t_k.$$

This holds for any $k \geq 1$. Therefore, because $t_k \to \infty$ as $k \to \infty$ we get $|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq T} \neq \emptyset\}| \neq o(T)$. Finally, this holds for any realization of $\mathbb{X}$ in the event $\mathcal{A} \cap \bigcap_{i \geq 1} \mathcal{E}_i \cap \bigcap_{i \geq 1} \mathcal{F}_i$. Therefore,

$$\mathbb{P}(|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq T} \neq \emptyset\}| = o(T)) \leq \mathbb{P}\left[\left(\mathcal{A} \cap \bigcap_{i \geq 1} \mathcal{E}_i \cap \bigcap_{i \geq 1} \mathcal{F}_i\right)^c\right] \leq 1 - \frac{\delta}{4} < 1.$$

Therefore, $\mathbb{X} \notin \text{SMV}_{([0,1],|\cdot|)}$ which contradicts the hypothesis. This concludes the proof that

$$\mathcal{L}_{\mathbb{X}}(4C1NN, \mathbb{1}_{\cdot \in A}) = 0 \quad (a.s.),$$

and hence, $\mathcal{S}_{\mathbb{X}}$ satisfies the $\sigma-$additivity property for disjoint sets.

Note that the choice of disjoint sets for the proof of $\sigma-$additivity was made for convenience so that the partition defined is not too complex. However to complete the proof of the $\sigma-$additivity of $\mathcal{S}_{\mathbb{X}}$, we have to prove that we can take unions of sets. Let $A_1, A_2 \in \mathcal{S}_{\mathbb{X}}$. We consider $A = A_1 \cup A_2$ and $f^*(\cdot) = \mathbb{1}_{\cdot \in A}$. Using the same arguments as above, we still have for $T \geq 1$,

$$\frac{1}{T} \sum_{t=1}^{T} (\mathbb{1}_{X_t \in A_i} \mathbb{1}_{X_{\phi(t)} \notin A} + \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A_i}) \to 0 \quad (a.s.).$$

for $i \in \{1, 2\}$. But note that for any $T \geq 1$,

$$\begin{aligned}
&\frac{1}{T} \sum_{t=1}^{T} \ell_{01}(4C1NN(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t), f^*(X_t)) \\
&\quad = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{X_t \in A} \mathbb{1}_{X_{\phi(t)} \notin A} + \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A} \\
&\quad \leq \frac{1}{T} \sum_{t=1}^{T} (\mathbb{1}_{X_t \in A_1} + \mathbb{1}_{X_t \in A_2}) \mathbb{1}_{X_{\phi(t)} \notin A} + \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{X_t \notin A} (\mathbb{1}_{X_{\phi(t)} \in A_1} + \mathbb{1}_{X_{\phi(t)} \in A_2}) \\
&\quad = \sum_{i=1}^{2} \frac{1}{T} \sum_{t=1}^{T} (\mathbb{1}_{X_t \in A_i} \mathbb{1}_{X_{\phi(t)} \notin A} + \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A_i}).
\end{aligned}$$

Therefore we obtain directly $\mathcal{L}_{\mathbb{X}}(4C1NN, \mathbb{1}_{\cdot \in A}) = 0 \quad (a.s.)$. This shows that $A_1 \cup A_2 \in \mathcal{S}_{\mathbb{X}}$ and ends the proof of the theorem. ■ As an immediate consequence of Theorem 3.5 and Proposition 3.1, we obtain the following results.

Theorem 3.6. *We have* $\text{SOUL}_{([0,1],|\cdot|),(\{0,1\},\ell_{01})} = \text{SMV}_{([0,1],|\cdot|)}$. *Further, in this setting for* $\mathcal{X} = [0, 1]$ *with usual measure, and for binary classification, 4C1NN is an optimistically universal learning rule.*

3.5.2 Generalization to standard Borel input spaces.

The specific choice of input space $\mathcal{X} = [0,1]$ was in fact not very restrictive. Indeed, any standard Borel input space $\mathcal{X}$ can be reduced to either $[0,1]$ or a countable set through the Kuratowski theorem. We recall that two standard Borel spaces i.e. complete separable Borel spaces, are Borel isomorphic if there exists a measurable bijection between them.

Theorem 3.7 (Kuratowski's theorem)**.** *Any standard Borel space $\mathcal{X}$ is Borel isomorphic to one of (1) $\mathbb{R}$, (2) $\mathbb{N}$ or (3) a finite space.*

This classical result can be found for example in [Kec12] (Section 15.B). Using this two reductions, we can generalize Theorem 3.6 to any standard Borel space $\mathcal{X}$.

Corollary 3.1. *For any standard Borel space $\mathcal{X}$ and binary classification, we have that $\textsc{Soul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} = SMV_{(\mathcal{X},\rho)}$. Further, there exists an optimistically universal learning rule.*

Proof The results are already known when $\mathcal{X}$ is countable and in these cases, memorization is an optimistically universal learning rule [Han21a]. We now fix a standard Borel space $\mathcal{X}$, Borel isomorphic to $\mathbb{R}$ and as a result Borel isomorphic to $[0,1]$. Let $g : \mathcal{X} \to [0,1]$ be a measurable bijection and a process $\mathbb{X} \in \text{SMV}_{(\mathcal{X},\rho)}$. Note that the process $g(\mathbb{X}) := (g(X_t))_{t\geq 1}$ belongs to $\text{SMV}_{([0,1],|\cdot|)}$ by bi-measurability of g. We can construct the learning rule $f_.$ for value setting $\mathcal{X}$ and binary classification such that for any $\boldsymbol{x}_{\leq t} \in \mathcal{X}^t$ and $\boldsymbol{y}_{<t} \in \{0,1\}^{t-1}$ we define $f_t(x_{<t}, y_{<t}, x_t) = 4C1NN_t(g(x_{<t}), y_{<t}, g(x_t))$. By construction, for target function $f^* : \mathcal{X} \to \{0,1\}$ this learning rule under $\mathbb{X}$ has same losses as 4C1NN under $g(\mathbb{X})$ for the target function $f^* \circ g^{-1}$. Therefore, $f_.$ is universally consistent under $\mathbb{X}$ which yields $\text{SMV}_{(\mathcal{X},\rho)} \subset \textsc{Soul}_{(\mathcal{X},\rho),(\{0,1\},\ell_{01})}$. Using Proposition 3.1 we have $\textsc{Soul}_{(\mathcal{X},\rho),(\{0,1\},\ell_{01})} = \text{SMV}_{(\mathcal{X},\rho)}$. We can also end the proof by noting that $f_.$ is an optimistically universal learning rule. ■

Although quite intuitive and direct, this generalization has two limitations. First, it only applies to standard Borel spaces instead of general separable Borel spaces. Second, it does not provide a practical optimistically universal rule in general. Indeed, the constructed optimistically universal learning rule in Corollary 3.1 uses a bimeasurable bijection between $\mathcal{X}$ and $[0,1]$—in the non-trivial case where $\mathcal{X}$ is Borel isomorphic to $\mathbb{R}$—which can be very complex and non-intuitive. For instance, the constructed learning rule for $[0,1]^2$ is not 4C1NN but instead a complex learning rule using a measurable bijection $[0,1] \to [0,1]^2$. In the next section we solve these two issues by showing that 2C1NN is optimistically universal in the general case.

3.6 Generalization to All Borel Context Spaces

In this section, we extend Corollary 3.1 to general binary classification, where $\mathcal{X}$ is a separable Borel space. This will then be used to prove the result for general output settings using a reduction technique described in Section 3.7. We show that 2C1NN is in fact always optimistically universal using the following proof structure. We first start with the case of binary classification $(\{0,1\}, \ell_{01})$. Specifically, we show that 2C1NN is universally consistent for binary classification under all processes in $\text{SMV}_{(\mathcal{X},\rho)}$ which yields

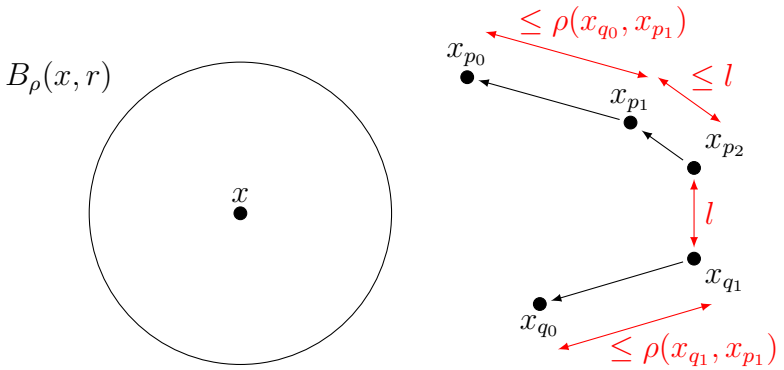

Figure 3.1: Illustration for Lemma 3.1 for $d = 2$ and $f = 1$, where the order of appearance is $p_0 < q_0 < p_1 < q_1 < p_2$. The arrows represent the relations of nodes within the tree $\mathcal{G}$, e.g., $\hat{Y}_{p_1} = Y_{p_0}$. If the end points x_{p_2} and x_{q_1} are close, then so are the beginning points x_{p_0} and x_{q_0}. The proof by induction is summarized by the upper bounds in red.

$\mathrm{SMV}_{(\mathcal{X},\rho)} \subset \textsc{Soul}_{(\mathcal{X},\rho),(\{0,1\},\ell_{01})}$. Together with Proposition 3.1, this shows that we have in fact an equality $\mathrm{SMV}_{(\mathcal{X},\rho)} = \textsc{Soul}_{(\mathcal{X},\rho),(\{0,1\},\ell_{01})}$ and as a result, that 2C1NN is optimistically universal.

We start by showing that under any process in $\mathrm{SMV}_{(\mathcal{X},\rho)}$, 2C1NN is consistent on functions representing balls of the metric ρ.

Proposition 3.3. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space constructed from the metric ρ. We consider the binary classification setting $\mathcal{Y} = \{0, 1\}$ and the ℓ_{01} binary loss. For any input process $\mathbb{X} \in SMV_{(\mathcal{X},\rho)}$, for any $x \in \mathcal{X}$, and $r > 0$, the learning rule 2C1NN is consistent for the target function $f^* = \mathbb{1}_{B_\rho(x,r)}$.*

To prove this result, we introduce a tree structure $\mathcal{G}$ for the 2C1NN algorithm on times t such that each new input is linked to its representant which was used to derive the target prediction. Times t corresponding to instances X_t that were previously visited are therefore not considered in this tree. Precisely, we consider parent relations given by $(t, \phi(t))$ for all times t such that a new input X_t was visited—i.e. memorization was not performed directly. By definition of the 2C1NN learning rule, no time $t \in \mathcal{G}$ has more than 2 children. Further, for any $t, t' \in \mathcal{G}$, if the time $t' < t$ is not present in dataset $\mathcal{D}_t$, it has exactly 2 children. The proof uses the following lemma.

Lemma 3.1. *Consider two distinct paths $p_d \to p_{d-1} \to \ldots \to p_1 \to p_0$ and $q_f \to q_{f-1} \to \ldots \to q_1 \to q_0$ i.e. $\phi(p_i) = p_{i-1}$ for $1 \leq i \leq d$ and $\phi(q_i) = q_{i-1}$ for $1 \leq i \leq f$. Suppose $p_0 < q_0$ and that there exists $t \geq \max(p_d, q_f)$ such that $p_d, q_f \in \mathcal{D}_t$ (in other words the two end times are in some final dataset). Then, with $v(0) := \max\{0 \leq i \leq d,\ p_i < q_0\}$ we have*

$$\rho(x_{p_{v(0)}}, x_{q_0}) \leq 2^{f+d+1}\rho(x_{p_d}, x_{q_f}) \quad \text{and} \quad \rho(x_{p_{v(0)}}, x_{p_d}) \leq 2^{f+d+1}\rho(x_{p_d}, x_{q_f}).$$

Proof Define

$$v(j) := \max\{0 \leq i \leq d,\ p_i < q_j\}, \quad j = 0, \ldots, f.$$
$$u(i) := \max\{0 \leq j \leq f,\ q_j < p_i\}, \quad i = v(0) + 1, \ldots, d,$$

Now observe that for any $v(0)+1 \leq i \leq d$, we have $q_{u(i)} \in \mathcal{D}_{p_i}$ i.e. the datapoint $q_{u(i)}$ is available in the current dataset. Indeed, it is possibly removed after all of its children have been revealed, in particular $q_{u(i)+1}$ if it exists. By definition of $u(i)$, even if $q_{u(i)+1}$ exists, it has not yet been revealed since $p_i < q_{u(i)+1}$. Therefore, we have $\rho(x_{p_i}, x_{p_{i-1}}) \leq \rho(x_{p_i}, x_{q_{u(i)}})$. Similarly, we have for all $1 \leq j \leq f$, $\rho(x_{q_j}, x_{q_{j-1}}) \leq \rho(x_{q_j}, x_{p_{v(j)}})$. We now take $v_0+1 \leq i < d$. We have $q_{u(i)} < p_{v(u(i))+1} < \ldots < p_i < q_{u(i)+1} < \ldots < q_{u(i+1)} < p_{i+1}$ (where some terms might not exist). Therefore,

$$\begin{aligned}
\rho(x_{p_i}, x_{q_{u(i)}}) &\leq \rho(x_{p_i}, x_{p_{i+1}}) + \rho(x_{p_{i+1}}, x_{q_{u(i+1)}}) + \rho(x_{q_{u(i)}}, x_{q_{u(i+1)}}) \\
&\leq 2\rho(x_{p_{i+1}}, x_{q_{u(i+1)}}) + \sum_{w=u(i)}^{u(i+1)-1} \rho(x_{q_w}, x_{q_{w+1}}) \\
&\leq 2\rho(x_{p_{i+1}}, x_{q_{u(i+1)}}) + \sum_{w=u(i)}^{u(i+1)-1} \rho(x_{p_i}, x_{q_{w+1}})
\end{aligned}$$

where in the last inequality, we used the fact that for all $u(i) \leq w \leq u(i+1)-1$, we have $v(w+1) = i$. Now observe that for any $u(i)+1 \leq w \leq u(i+1)-1$,

$$\rho(x_{p_i}, x_{q_w}) \leq \rho(x_{p_i}, x_{q_{w+1}}) + \rho(x_{q_w}, x_{q_{w+1}}) \leq 2\rho(x_{p_i}, x_{q_{w+1}}).$$

Therefore we have by induction $\rho(x_{p_i}, x_{q_w}) \leq 2^{u(i+1)-w}\rho(x_{p_i}, x_{q_{u(i+1)}})$. which yields

$$\rho(x_{p_i}, x_{q_{u(i)}}) \leq 2\rho(x_{p_{i+1}}, x_{q_{u(i+1)}}) + (2^{u(i+1)-u(i)} - 1)\rho(x_{p_i}, x_{q_{u(i+1)}}).$$

Finally, we observe that $\rho(x_{p_i}, x_{q_{u(i+1)}}) \leq \rho(x_{p_i}, x_{p_{i+1}}) + \rho(x_{p_{i+1}}, x_{q_{u(i+1)}}) \leq 2\rho(x_{p_{i+1}}, x_{q_{u(i+1)}})$. Hence,

$$\rho(x_{p_i}, x_{q_{u(i)}}) \leq 2^{u(i+1)-u(i)+1}\rho(x_{p_{i+1}}, x_{q_{u(i+1)}}).$$

By recursion, this yields

$$\rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}}) \leq 2^{u(d)-u(v(0)+1)+(d-v(0)-1)}\rho(x_{p_d}, x_{q_{u(d)}}).$$

We now relate the quantity $\rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}})$ (resp. $\rho(x_{p_d}, x_{q_{u(d)}})$) to $\rho(x_{p_{v(0)}}, x_{q_0})$ (resp. $\rho(x_{p_d}, x_{q_d})$). We have by construction $p_{v(0)} < q_0 < q_1 < \ldots < q_{u(v(0)+1)} < p_{v(0)+1}$. Therefore, similarly to before,

$$\begin{aligned}
\rho(x_{p_{v(0)}}, x_{q_0}) &\leq \rho(x_{p_{v(0)}}, x_{p_{v(0)+1}}) + \rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}}) + \sum_{w=0}^{u(v(0)+1)-1} \rho(x_{q_w}, x_{q_{w+1}}) \\
&\leq 2\rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}}) + \sum_{w=1}^{u(v(0)+1)} \rho(x_{p_{v(0)}}, x_{q_w}).
\end{aligned}$$

But $\rho(x_{p_{v(0)}}, x_{q_w}) \leq \rho(x_{p_{v(0)}}, x_{q_{w+1}}) + \rho(x_{q_w}, x_{q_{w+1}}) \leq 2\rho(x_{p_{v(0)}}, x_{q_{w+1}})$. Hence $\rho(x_{p_{v(0)}}, x_{q_w}) \leq 2^{u(v(0)+1)-w}\rho(x_{p_{v(0)}}, x_{q_{u(v(0)+1)}}) \leq 2^{u(v(0)+1)-w+1}\rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}})$. Then,

$$\rho(x_{p_{v(0)}}, x_{q_0}) \leq 2^{u(v(0)+1)+1}\rho(x_{p_{v(0)+1}}, x_{q_{u(v(0)+1)}}) \leq 2^{u(d)+(d-v(0))}\rho(x_{p_d}, x_{q_{u(d)}}).$$

Finally, we have $q_{u(d)} < p_d < q_{u(d)+1} < \ldots < q_f$. Then,

$$\begin{aligned}\rho(x_{p_d}, x_{q_{u(d)}}) &\leq \sum_{w=u(d)}^{f-1} \rho(x_{q_w}, x_{q_{w+1}}) + \rho(x_{p_d}, x_{q_f}) \\ &\leq \sum_{w=u(d)}^{f-1} \rho(x_{p_d}, x_{q_{w+1}}) + \rho(x_{p_d}, x_{q_f}).\end{aligned}$$

Again, note that for $u(d)+1 \leq w \leq f-2$, we have $\rho(x_{p_d}, x_{q_w}) \leq \rho(x_{p_d}, x_{q_{w+1}}) + \rho(x_{q_w}, x_{q_{w+1}}) \leq 2\rho(x_{p_d}, x_{q_{w+1}})$. Hence, $\rho(x_{p_d}, x_{q_w}) \leq 2^{f-w}\rho(x_{p_d}, x_{q_f})$ and we obtain

$$\rho(x_{p_d}, x_{q_{u(d)}}) \leq (2^{f-u(d)} - 1)\rho(x_{p_d}, x_{q_f}) + \rho(x_{p_d}, x_{q_f}) = 2^{f-u(d)}\rho(x_{p_d}, x_{q_f}).$$

Putting everything together yields

$$\rho(x_{p_{v(0)}}, x_{q_0}) \leq 2^{f+d}\rho(x_{p_d}, x_{q_f}).$$

Finally, we compute

$$\begin{aligned}\rho(x_{p_{v(0)}}, x_{p_d}) &\leq \sum_{i=v(0)+1}^{d} \rho(x_{p_{i-1}}, x_{p_i}) \\ &\leq \sum_{i=v(0)+1}^{d} \rho(x_{p_i}, x_{q_{u(i)}}) \\ &\leq \sum_{i=v(0)+1}^{d} 2^{u(d)-u(i)+d-i}\rho(x_{p_d}, x_{q_{u(d)}}) \\ &\leq \sum_{i=v(0)+1}^{d} 2^{u(d)-u(v(0)+1)+d-i}\rho(x_{p_d}, x_{q_{u(d)}}) \\ &\leq 2^{u(d)-u(v(0)+1)+d-v(0)}\rho(x_{p_d}, x_{q_{u(d)}}) \\ &\leq 2^{f-u(v(0)+1)+d-v(0)}\rho(x_{p_d}, x_{q_f}) \\ &\leq 2^{f+d}\rho(x_{p_d}, x_{q_f}).\end{aligned}$$

This ends the proof of the lemma. ■

We are now ready to show that 2C1NN is consistent on functions representing balls of the metric ρ, under any process in $\text{SMV}_{(\mathcal{X},\rho)}$.

Proof of Proposition 3.3 We fix $\bar{x} \in \mathcal{X}$, $r > 0$ and $f^* = \mathbb{1}_{B(\bar{x},r)}$. We reason by the contrapositive and suppose that 2C1NN is not consistent on f^*. We will show that the process $\mathbb{X}$ disproves the $\text{SMV}_{(\mathcal{X},\rho)}$ condition by considering a partition for which, the process $\mathbb{X}$ does not visit a sublinear number of sets with nonzero probability.

Because 2C1NN is not consistent, $\delta := \mathbb{P}(\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > 0) > 0$. Therefore, there exists $0 < \epsilon \leq 1$ such that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > \epsilon) > \frac{\delta}{2}$. Denote $\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > \epsilon\}$.

We therefore have $\mathbb{P}(\mathcal{A}) > \frac{\delta}{2}$. We now define a partition $\mathcal{P}$. Because $\mathcal{X}$ is separable, there exists a sequence $(x^i)_{i\geq 1}$ of elements of $\mathcal{X}$ which is dense i.e.

$$\forall x \in \mathcal{X}, \quad \inf_{i\geq 1} \rho(x, x^i) = 0.$$

We focus for now on the sphere $S(\bar{x}, r)$ and for any $\tau > 0$ we take $(P_i(\tau))_{i\geq 1}$ the sequence of sets included in $S(\bar{x}, r)$ defined by

$$P_i(\tau) := \big(S(\bar{x}, r) \cap B(x^i, \tau)\big) \setminus \left(\bigcup_{1\leq j<i} B(x^j, \tau)\right).$$

These sets are disjoint. Further, they partition $S(\bar{x}, r)$. Indeed, if $x \in S(\bar{x}, r)$, let $i \geq 1$ such that $\rho(x, x^i) \leq \tau$. Then, $x \in S(\bar{x}, r) \cap B(x^i, \tau) \subset \bigcup_{j\leq i} P_j^\tau$. We now pose

$$\tau_l := c_\epsilon \cdot \frac{r}{2^{l+1}},$$

for $l \geq 1$, where $c_\epsilon := \frac{1}{2\cdot 2^{2^5/\epsilon}}$ is a constant dependant on ϵ only. We also pose $\tau_0 = r$. Then, because $\mathbb{X} \in \mathrm{SMV}_{(\mathcal{X},\rho)}$, the process visits a sublinear number of sets of $\mathcal{P}_i(\tau_l)$ almost surely. Therefore, there exists an increasing sequence $(n_l)_{l\geq 1}$ such that for any $l \geq 1$,

$$\mathbb{P}\left[\forall n \geq n_l,\ |\{i,\ P_i(\tau_l) \cap \mathbb{X}_{<n} \neq \emptyset\}| \leq \frac{\epsilon}{2^7} n\right] \geq 1 - \frac{\delta}{2 \cdot 2^{l+2}} \quad \text{and} \quad n_{l+1} \geq \frac{2^6}{\epsilon} n_l$$

We denote by $\mathcal{E}_l$ this event. Thus, $\mathbb{P}[\mathcal{E}_l] \leq \frac{\delta}{2\cdot 2^{l+2}}$. Now, for any $l \geq 1$, we now construct $\mu_l > 0$ such that

$$\mathbb{P}\left[\min_{i<j\leq n_l,\ X_i \neq X_j} \rho(X_i, X_j) > \mu_l\right] \geq 1 - \frac{\delta}{2 \cdot 2^{l+2}}.$$

We denote this event by $\mathcal{F}_l$. Thus $\mathbb{P}[\mathcal{F}_l] \leq \frac{\delta}{2\cdot 2^{l+2}}$. Note that the sequence $(\mu_l)_{l\geq 1}$ is non-increasing. We now define radiuses $(z^i)_{i\geq 1}$ as follows:

$$z^i = \begin{cases} \mu_{l_i+1} & \text{if } \rho(x^i, \bar{x}) < r, \text{ where } \frac{r}{2^{l_i+1}} < r - \rho(x^i, \bar{x}) \leq \frac{r}{2^{l_i}} \\ 0 & \text{if } \rho(x^i, \bar{x}) \geq r, \end{cases}$$

and consider the sets $R_i := B(x^i, z^i) \cap \left\{x \in \mathcal{X} :\ \rho(x, \bar{x}) < r - \frac{r}{2^{l_i+2}}\right\}$. We construct

$$P_i := R_i \setminus \left(\bigcup_{k<i} R_k\right),$$

for $i \geq 1$. As shown in the following lemma, $(P_i)_{i\geq 1}$ forms a partition of $B(\bar{x}, r)$.

Lemma 3.2. *$(P_i)_{i\geq 1}$ forms a partition of $B(\bar{x}, r)$.*

We now define a second partition. We start by defining a sequence of radiuses $(r^i)_{i\geq 1}$ as follows

$$r^i = \begin{cases} c_\epsilon \inf\limits_{x:\ \rho(x,\bar{x})\leq r} \rho(x^i, x) & \text{if } \rho(x^i, \bar{x}) > r, \\ c_\epsilon \inf\limits_{x:\ \rho(x,\bar{x})\geq r} \rho(x^i, x) & \text{if } \rho(x^i, \bar{x}) < r, \\ 0 & \text{if } \rho(x^i, \bar{x}) = r. \end{cases}$$

We consider the sets $(A_i)_{i\geq 0}$ given by $A_0 = S(\bar{x}, r)$ and for $i \geq 1$,

$$A_i = B(x^i, r^i) \setminus \left(\bigcup_{1\leq j<i} B(x^j, r^j) \right).$$

We now show that these sets form a partition in the following lemma.

Lemma 3.3. *$(A_i)_{i\geq 0}$ forms a partition of $\mathcal{X}$.*

We now formally consider the product partition of $(P_i)_{i\geq 1}$ and $(A_i)_{i\geq 0}$ i.e.

$$\mathcal{Q}: \quad \bigcup_{i\geq 0,\ A_i\subset B(\bar{x},r)} \bigcup_{j\geq 1} (A_i \cap P_j) \cup \bigcup_{i\geq 0,\ A_i\subset \mathcal{X}\setminus B(\bar{x},r)} A_i.$$

where we used the fact that sets A_i satisfy either $A_i \subset B(\bar{x}, r)$ or $A_i \subset \mathcal{X} \setminus B(\bar{x}, r)$. We will show that this partition disproves the $\text{SMV}_{(\mathcal{X},\rho)}$ hypothesis on $\mathbb{X}$. In practice, we will either prove that the process visits many sets from partition $(A_i)_{i\geq 0}$ or $(P_i)_{i\geq 1}$ and use the fact that the same analysis would work for $\mathcal{Q}$, the product partition as well.

We now consider a specific realization $\boldsymbol{x} = (x_t)_{t\geq 0}$ of the process $\mathbb{X}$ falling in the event $\mathcal{A} \bigcap_{l\geq 1} (\mathcal{E}_l \cap \mathcal{F}_l)$. This event has probability

$$\mathbb{P}\left[\mathcal{A} \bigcap_{l\geq 1} (\mathcal{E}_l \cap \mathcal{F}_l)\right] \geq \mathbb{P}[\mathcal{A}] - \sum_{l\geq 1} (\mathbb{P}[\mathcal{E}_l^c] + \mathbb{P}[\mathcal{F}_l^c]) \geq \frac{\delta}{2} - \frac{\delta}{4} = \frac{\delta}{4}.$$

Note that $\boldsymbol{x}$ is not random anymore. We now show that $\boldsymbol{x}$ does not visit a sublinear number of sets in the partition $\mathcal{Q}$.

We now denote by $(t_k)_{k\geq 1}$ the increasing sequence of all times when 2C1NN makes an error in the prediction of $f^*(x_t)$. Because the event $\mathcal{A}$ is satisfied, $\mathcal{L}_{\boldsymbol{x}}(2C1NN, f^*) > \epsilon$, therefore, we can define an increasing sequence of times $(T_l)_{l\geq 1}$ such that

$$\frac{1}{T_l} \sum_{t=1}^{T_l} \ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) > \frac{\epsilon}{2}.$$

For any $l \geq 1$ consider the last index $k = \max\{u, t_u \leq T_l\}$ when 2C1NN makes a mistake. Then we obtain $k > \frac{\epsilon}{2} T_l \geq \frac{\epsilon}{2} t_k$. Considering the fact that $(T_l)_{l\geq 1}$ is an increasing unbounded sequence we therefore obtain an increasing sequence of indices $(k_l)_{l\geq 1}$ such that $t_{k_l} < \frac{2k_l}{\epsilon}$.

At an iteration where the new input x_t has not been previously visited we will denote by $\phi(t)$ the index of the nearest neighbor of the current dataset in the 2C1NN learning rule. Now let $l \geq 1$. We focus on the time t_{k_l}. Consider the tree $\mathcal{G}$ where nodes are times $\mathcal{T} := \{t,\ t \leq t_{k_l},\ x_t \notin \{x_u, u < t\}\}$ for which a new input was visited, where the parent relations are given by $(t, \phi(t))$ for $t \in \mathcal{T} \setminus \{1\}$. In other words, we construct the tree in which a new input is linked to its representant which was used to derive the target prediction. Note that by definition of the 2C1NN learning rule, each node has at most 2 children and a node is not in the dataset at time t_{k_l} when it has exactly 2 children.

Step 1. We now suppose that the majority of input points on which 2C1NN made a mistake belong to $B(\bar{x}, r)$ i.e.

$$|\{t \leq t_{k_l},\ \ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) = 1,\ x_t \in B(\bar{x}, r)\}| \geq \frac{k_l}{2},$$

or equivalently $|\{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}| \geq \frac{k_l}{2}$.

Let us now consider the subgraph $\tilde{\mathcal{G}}$ given by restricting $\mathcal{G}$ only to nodes in the ball $B(\bar{x}, r)$—which are mapped to the true value 1—i.e. on times $\{t \in \mathcal{T},\ x_t \in B(\bar{x}, r)\}$. In this subgraph, the only times with no parent are times t_k with $k \leq k_l$ and $x_{t_k} \in B(\bar{x}, r)$ and possibly time $t = 1$. Indeed, if a time in $\tilde{\mathcal{G}}$ has a parent $\phi(t)$ in $\tilde{\mathcal{G}}$, the prediction of 2C1NN for x_t returned the correct answer 1. The converse is also true except for the root time $t = 1$ which has no parent in $\mathcal{G}$. Therefore, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with roots times $\{t_k,\ k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}$—and possibly $t = 1$ if $x_1 \in B(\bar{x}, r)$. For a given time t_k with $k \leq k_l$ and $x_{t_k} \in B(\bar{x}, r)$, we will denote by $\mathcal{T}_k$ the corresponding tree in $\tilde{\mathcal{G}}$ with root t_k. We will say that the $\mathcal{T}_k$ is a *good* tree if all times $t \in \mathcal{T}_k$ of this tree are parent in $\mathcal{G}$ to at most 1 time from $\mathcal{X} \setminus B(\bar{x}, r)$ i.e. if

$$\forall t \in \mathcal{T}_k, \quad |\{u \leq t_{k_l},\ \phi(u) = t,\ \rho(x_u, \bar{x}) \geq r\}| \leq 1.$$

We denote by $G = \{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r),\ \mathcal{T}_k \text{ good}\}$ the set of indices of good trees. By opposition, we will say that a tree is *bad* otherwise. We now give a simple upper bound on N_{bad} the number of bad trees. Note that for any $t \in \mathcal{T}_k$, times in $\{u \leq t_{k_l},\ \phi(u) = t,\ \rho(x_u, \bar{x}) \geq r\}$ are times when 2C1NN makes a mistake on $\mathcal{X} \setminus B(\bar{x}, r)$. Therefore,

$$\sum_{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)} \sum_{t \in \mathcal{T}_k} |\{u < t_{k_l},\ \phi(u) = t,\ \rho(x_u, \bar{x}) \geq r\}| \leq |\{k \leq t_{k_l}, \rho(x_{t_k}, \bar{x}) \geq r\}| \leq \frac{k_l}{2}$$

because by hypothesis $|\{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}| \geq \frac{k_l}{2}$. Therefore, since each bad tree contains a node which is parent to at least 2 times of mistake in $\mathcal{X} \setminus B(\bar{x}, r)$, we obtain

$$N_{\text{bad}} \leq \sum_{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)} \sum_{t \in \mathcal{T}_k} \frac{1}{2} |\{u < t_{k_l},\ \phi(u) = t,\ \rho(x_u, \bar{x}) \geq r\}| \leq \frac{k_l}{4}.$$

Thus, the number of good trees is $|G| \geq |\{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}| - N_{\text{bad}} \geq \frac{k_l}{4}$. Now note that trees are disjoint, therefore, $\sum_{k \in G} |\mathcal{T}_k| \leq t_{k_l} < \frac{2k_l}{\epsilon}$. Therefore,

$$\sum_{k \in G} \mathbb{1}_{|\mathcal{T}_k| \leq \frac{16}{\epsilon}} = |G| - \sum_{k \in G} \mathbb{1}_{|\mathcal{T}_k| > \frac{16}{\epsilon}} > |G| - \frac{\epsilon}{16} \sum_{k \in G} |\mathcal{T}_k| \geq \frac{k_l}{8}.$$

We will say that a tree $|\mathcal{T}_k|$ is *sparse* if it is good and has at most $\frac{\epsilon}{16}$ nodes. With $S := \{k \in G,\ |\mathcal{T}_k| \leq \frac{16}{\epsilon}\}$ the set of sparse trees, the above equation we have $|S| \geq \frac{k_l}{8}$. We now focus only on sparse trees $\mathcal{T}_k$ for $k \in S$ and analyze their relation with the final dataset $\mathcal{D}_{t_{k_l}}$. Precisely, for a sparse tree $\mathcal{T}_k$, denote $\mathcal{V}_k = \mathcal{T}_k \cap \mathcal{D}_{t_{k_l}}$ the set of times which are present in the final dataset and belong to the tree induced by error time t_k. Because each node of $\mathcal{T}_k$ and not present in $\mathcal{D}_{t_{k_l}}$ has at least 1 children in $\mathcal{T}$, we note that $\mathcal{V}_k \neq \emptyset$. We now consider

the path from a node of $\mathcal{V}_k$ to the root t_k. We denote by $d(k)$ the depth of this node in $\mathcal{V}_k$ and denote the path by $p^k_{d(k)} \to p^k_{d(k)-1} \to p^k_0 = t_k$ where $p^k_{d(k)} \in \mathcal{V}_k$. Then we have,

$$d(k) \leq |\mathcal{T}_k| - 1 \leq \frac{16}{\epsilon} - 1.$$

Each arc of this path represents the fact that at the corresponding iteration p^k_i of 2C1NN, the parent $x_{p^k_{i-1}}$ was closer from $x_{p^k_i}$ than any other point of the current dataset $\mathcal{D}_{p^k_i}$. We will now show that all the points $\{p^k_{d(k)},\ k \in S\}$ fall in distinct sets of the partition $(A_i)_{i\geq 0}$. Suppose by contradiction that we have $k_1 \neq k_2 \in S$ falling into the same set A_i. Note that because $x_{p^{k_1}_{d(k_1)}}, x_{p^{k_2}_{d(k_2)}} \in B(\bar{x}, r)$, we obtain $A_i \cap B(\bar{x}, r) \neq \emptyset$. However, the partition $(A_i)_{i\geq 0}$ was constructed so that sets are included totally in either $B(\bar{x}, r)$, $S(\bar{x}, r)$ or $\{x \in \mathcal{X},\ \rho(x, \bar{x}) > r\}$. Therefore, we obtain $A_i \subset B(\bar{x}, r)$ and $x^i \in B(\bar{x}, r)$. We can now apply Lemma 3.1 to $p^{k_1}_{d(k_1)} \to p^{k_1}_{d(k_1)-1} \to \ldots \to p^{k_1}_0$ and $p^{k_2}_{d(k_2)} \to p^{k_2}_{d(k_2)-1} \to \ldots \to p^{k_2}_0$—which we write by convenience $p_d \to p_{d-1} \to \ldots \to p_1 \to p_0$ and $q_f \to q_{f-1} \to \ldots \to q_1 \to q_0$—assuming without loss of generality that $p_0 < q_0$. Therefore, $\rho(x_{p_{v(0)}}, x_{q_0}) \leq 2^{f+d}\rho(x_{p_d}, x_{q_f}) \leq 2^{f+d+1}r^i$ and $\rho(x_{p_{v(0)}}, x_{p_d}) \leq 2^{f+d}\rho(x_{p_d}, x_{q_f}) \leq 2^{f+d+1}r^i$. But recall that these two paths come from sparse trees, so $d, f \leq \frac{16}{\epsilon} - 1$. Hence, $2^{f+d+1} \leq \frac{1}{2}2^{2^5/\epsilon} = \frac{1}{4c_\epsilon}$. Let us now consider $x_{\phi(q_0)}$ the point which induced a mistake in the prediction of x_{q_0}, i.e. $\rho(x_{\phi(q_0)}, \bar{x}) \geq r$. Then,

$$\begin{aligned}
\rho(x_{q_0}, x_{\phi(q_0)}) &\geq \rho(x_{\phi(q_0)}, x^i) - \rho(x^i, x_{p_d}) - \rho(x_{p_d}, x_{p_{v(0)}}) - \rho(x_{p_{v(0)}}, x_{q_0}) \\
&\geq \frac{r^i}{c_\epsilon} - r^i - \frac{r^i}{4c_\epsilon} - \frac{r^i}{4c_\epsilon} \\
&\geq \frac{r^i}{4c_\epsilon}
\end{aligned}$$

where in the last inequality we used the fact that $c_\epsilon < \frac{1}{4}$. Recall that we also proved $\rho(x_{p_{v(0)}}, x_{q_0}) \leq \frac{r^i}{4c_\epsilon} < \rho(x_{q_0}, x_{\phi(q_0)})$. However, datapoint $x_{p_{v(0)}}$ is available in dataset $\mathcal{D}_{q_0}$. This contradicts the fact that $x_{\phi(t)}$ was chosen as representant for x_{q_0}. This ends the proof that all the points $\{p^k_{d(k)},\ k \in S\}$ fall in distinct sets of the partition $(A_i)_{i\geq 0}$. Therefore,

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |S| \geq \frac{k_l}{8} \geq \frac{\epsilon}{16} t_{k_l}.$$

Step 2. We now turn to the case when the majority of input points on which 2C1NN made a mistake are not in the ball $B(\bar{x}, r)$ i.e.

$$|\{t \leq t_{k_l},\ \ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) = 1,\ \rho(x_t, \bar{x}) \geq r\}| \geq \frac{k_l}{2},$$

or equivalently $|\{k \leq k_l,\ \rho(x_{t_k}, \bar{x}) \geq r\}| \geq \frac{k_l}{2}$. Similarly as the previous case, we consider the graph $\tilde{G}$ given by restricting $\mathcal{G}$ only to nodes outside the ball $B(\bar{x}, r)$ i.e. on times $\{t \in \mathcal{T}, \rho(x_t, \bar{x}) \geq r\}$. Again, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with root times $\{t_k,\ k \leq k_l,\ \rho(x_{t_k}, \bar{x}) \geq r\}$ (and possibly $t = 1$). We denote $\mathcal{T}_k$ the corresponding tree of $\tilde{\mathcal{G}}$ rooted in t_k. Similarly to above, a tree is *sparse* if

$$\forall t \in \mathcal{T}_k, \quad |\{u \leq t_{k_l},\ \phi(u) = t,\ \rho(x_u, \bar{x}) < r\}| \leq 1 \quad \text{and} \quad |\mathcal{T}_k| \leq \frac{16}{\epsilon}.$$

If $S = \{k \leq k_l, ; \rho(x_{t_k}, \bar{x}) \geq r,\ \mathcal{T}_k \text{ sparse}\}$ denotes the set of sparse trees, the same proof as above shows that $|S| \geq \frac{k_l}{8}$. Again, for any $k \in S$, if $d(k)$ denotes the depth of some node from $\mathcal{V}_k := \mathcal{T}_k \cap \mathcal{D}_{t_{k_l}}$ in $\mathcal{T}_k$ we have $d(k) \leq \frac{16}{\epsilon} - 1$. For each $k \in S$ we consider the path from this node of $\mathcal{V}_k$ to the root t_k: $p^k_{d(k)} \to p^k_{d(k)-1} \to \ldots \to p^k_0 = t_k$ where $p^k_{d(k)} \in \mathcal{V}_k$. The same proof as above shows that all the points $\{p^k_{d(k)},\ k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}$ lie in distinct sets of the partition $(A_i)_{i \geq 0}$.

Indeed, let $p_d \to p_{d-1} \to \ldots \to p_1 \to p_0$ and $q_f \to q_{f-1} \to \ldots \to q_1 \to q_0$ two such paths with $\rho(x_{p_d}, \bar{x}) > r$ and $\rho(x_{q_f}, \bar{x}) > r$ and suppose by contradiction that $x_{p_d}, x_{q_f} \in A_i$ for some $i \geq 0$. Necessarily, $i \geq 1$ and $\rho(x^i, \bar{x}) > r$. Lemma 3.1 gives again $\rho(x_{p_{v(0)}}, x_{q_0}), \rho(x_{p_{v(0)}}, x_{p_d}) \leq 2^{f+d}\rho(x_{p_d}, x_{q_f}) \leq 2^{f+d+1} r^i \leq \frac{r^i}{4c_\epsilon}$. Then, if $x_{\phi(q_0)}$ is the point that induced a mistake in the prediction of x_{q_0}, we have $\rho(x_{\phi(q_0)}, \bar{x}) < r$. Using the definition of r^i we obtain the same computations

$$\rho(x_{q_0}, x_{\phi(q_0)}) \geq \rho(x_{\phi(q_0)}, x^i) - \rho(x^i, x_{p_d}) - \rho(x_{p_d}, x_{p_{v(0)}}) - \rho(x_{p_{v(0)}}, x_{q_0}) \geq \frac{r^i}{4c_\epsilon} > \rho(x_{p_{v(0)}}, x_{q_0})$$

which contradicts the fact that $x_{\phi(q_0)}$ was used as representant for x_{q_0}. This ends the proof that all the points $\{p^k_{d(k)},\ k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}$ lie in distinct sets of the partition $(A_i)_{i \geq 0}$. Suppose $|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2}$, then we have

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l}{16} \geq \frac{\epsilon}{32} t_{k_l}.$$

Step 3. In this last step, we suppose again that the majority of input points on which 2C1NN made a mistake are not in the ball $B(\bar{x}, r)$ and that $|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| < \frac{|S|}{2}$. Therefore, we obtain

$$|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}| = |S| - |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l}{16} \geq \frac{\epsilon}{32} t_{k_l}.$$

We will now make use of the partition $(P_i)_{i \geq 1}$. Because $(n_u)_{u \geq 1}$ is an increasing sequence, let $u \geq 1$ such that $n_{u+1} \leq t_{k_l} \leq n_{u+2}$ (we can suppose without loss of generality that $t_{k_0} > n_2$). Note that we have $n_u \leq \frac{\epsilon}{2^6} n_{u+1} \leq \frac{\epsilon}{2^6} t_{k_l}$. Let us now analyze the process between times n_u and t_{k_l}. In particular, we are interested in the indices $T = \{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}$ and times $\mathcal{U}_u = \{p^k_{d(k)} :\ n_u < p^k_{d(k)} \leq k_l,\ k \in T\}$. In particular, we have

$$|\mathcal{U}_u| \geq |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}| - n_u \geq \frac{\epsilon}{32} t_{k_l} - \frac{\epsilon}{2^6} t_{k_l} = \frac{\epsilon}{2^6} t_{k_l}.$$

Because the event $\mathcal{E}_u$ is met, we have

$$|\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\mathcal{U}_u} \neq \emptyset\}| \leq |\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \leq \frac{\epsilon}{2^7} t_{k_l}.$$

Note that $\boldsymbol{x}_{\mathcal{U}_u} \subset S(\bar{x}, r)$. Therefore, each of the points in $\boldsymbol{x}_{\mathcal{U}_u}$ falls into one of the sets $(P_i(\tau_u))_{i \geq 1}$. Let $i \geq 1$ such that the set $P_i(\tau_u)$ was visited by $\boldsymbol{x}_{\mathcal{U}_u}$ and consider $T_i = \{k \in$

$T,\ x_{p^k_{d(k)}} \in A_i\}$. We will show that at least $|T_i| - 1$ of the points $\{x_{\phi(t_k)},\ k \in T_i\}$ fall in $B(\bar{x}, r) \setminus B(\bar{x}, r - \frac{r}{2^{u+2}})$.

To do so, let $k_1, k_2 \in T_i$. Similarly as above, for simplicity, we will refer to the path $p^{k_1}_{d(k_1)} \to p^{k_1}_{d(k_1)-1} \to \ldots \to p^{k_1}_0$ (resp. $p^{k_2}_{d(k_2)} \to p^{k_2}_{d(k_2)-1} \to \ldots \to p^{k_2}_0$) as $p_d \to p_{d-1} \to \ldots \to p_1 \to p_0$ (resp. $q_f \to q_{f-1} \to \ldots \to q_1 \to q_0$), and assume without loss of generality that $p_0 < q_0$. Note that by hypothesis, $k_1, k_2 \in T_i$, therefore, $\rho(x_{p_d}, x^i), \rho(x_{q_f}, x^i) \leq \tau_u$ Then, using the above computations yields

$$\rho(x_{p_{v(0)}}, x_{q_0}) \leq 2^{f+d}\rho(x_{p_d}, x_{q_f}) \leq 2^{f+d}(\rho(x_{p_d}, x^i) + \rho(x_{q_f}, x^i)) \leq 2^{f+d+1}\tau_u \leq \frac{\tau_u}{4c_\epsilon},$$

where in the last inequality we used the fact that $f, d \leq \frac{16}{\epsilon} - 1$ hence $2^{f+d+1} \leq \frac{1}{4c_\epsilon}$. Now by definition of a representant, we obtain

$$\rho(x_{\phi(q_0)}, x_{q_0}) \leq \rho(x_{p_{v(0)}}, x_{q_0}) \leq \frac{r}{8 \cdot 2^u}.$$

Therefore, $\rho(x_{\phi(q_0)}, \bar{x}) \geq \rho(x_{q_0}, \bar{x}) - \rho(x_{\phi(q_0)}, x_{q_0}) \geq r - \frac{r}{8 \cdot 2^u}$. Because $x_{\phi(q_0)}$ induced a mistake in the prediction for x_{q_0} we have $x_{\phi(q_0)} \in B(\bar{x}, r)$. Now order $T_i = \{k_1 < \ldots < k_{|T_i|}\}$. We then have $t_{k_1} < \ldots < t_{k_{|T_i|}}$. The argument above then shows that for any $2 \leq j \leq |T_i|$, we have $x_{\phi(t_{k_j})} \in B(\bar{x}, r) \setminus B(\bar{x}, r - \frac{r}{2^{u+3}})$. Therefore, defining $T' := \{k \in T,\ r - \frac{r}{2^{u+3}} \leq \rho(x_{\phi(t_k)}, \bar{x}) < r\}$ we obtain

$$|T'| \geq |\mathcal{U}_u| - |\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\mathcal{U}_u} \neq \emptyset\}| \geq \frac{\epsilon}{2^7} t_{k_l}.$$

We will now show that all the points in $\{x_{t_k},\ k \in T'\}$ lie in distinct sets of $(P_i)_{i \geq 1}$. Note that because we have $t_{k_l} \leq n_{u+2}$ and because the event $\mathcal{F}_{u+2}$ is met, we have that for any $p, q \in T'$ that $\rho(x_{\phi(t_p)}, x_{\phi(t_q)}) > \mu_{u+2}$. Now suppose by contradiction that $x_{\phi(t_p)}, x_{\phi(t_q)} \in P_i$ for some $i \geq 1$. Then, with l_i such that $r - \frac{r}{2^{l_i}} \leq \rho(x^i, \bar{x}) < r - \frac{r}{2^{l_i+1}}$ we have that

$$x_{\phi(t_p)}, x_{\phi(t_q)} \in \left\{x \in \mathcal{X} :\ \rho(x, \bar{x}) < r - \frac{r}{2^{l_i+2}}\right\}$$

But we know that $\rho(x_{\phi(t_p)}, \bar{x}) \geq r - \frac{r}{2^{u+3}}$. Therefore we obtain $r - \frac{r}{2^{l_i+2}} > r - \frac{r}{2^{u+3}}$ and hence $l_i \geq u + 1$. Recall that $P_i \subset B(x^i, \mu_{l_i+1})$. Therefore, we obtain

$$\rho(x_{\phi(t_p)}, x_{\phi(t_q)}) \leq \mu_{l_i+1} \leq \mu_{u+2},$$

which contradicts the fact that $\rho(x_{t_p}, x_{t_q}) > \mu_{u+2}$. This ends the proof that all points of $\{x_{t_k},\ k \in T'\}$ lie in distinct subsets of $(P_i)_{i \geq 1}$. Now we obtain

$$|\{i,\ P_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |T'| \geq \frac{\epsilon}{2^7} t_{k_l}.$$

Step 4. In conclusion, in all cases, we obtain

$$|\{Q \in \mathcal{Q},\ Q \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq \max(|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}|, |\{i,\ P_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}|) \geq \frac{\epsilon}{2^7} t_{k_l}.$$

Because this is true for all $l \geq 1$ and t_{k_l} is an increasing sequence, we conclude that $\boldsymbol{x}$ disproves the $\text{SMV}_{(\mathcal{X},\rho)}$ condition for $\mathcal{Q}$. Recall that this holds whenever the event $\mathcal{A} \bigcap_{l\geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)$ is met. Thus,

$$\mathbb{P}[|\{Q \in \mathcal{Q},\ Q \cap \mathbb{X}_{<T}\}| = o(T)] \leq 1 - \mathbb{P}[\mathcal{A} \bigcap_{l\geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)] \leq 1 - \frac{\delta}{4} < 1.$$

This shows that $\mathbb{X} \notin \text{SMV}_{(\mathcal{X},\rho)}$ which is absurd. Therefore 2C1NN is consistent on f^*. This ends the proof of the proposition. ■

We can now show that 2C1NN is optimistically universal for the binary classification setting, with a similar proof structure to Theorem 3.5. Precisely, we show that under any process $\mathbb{X} \in \text{SMV}_{(\mathcal{X},\rho)}$, the functions on which it is consistent form a $\sigma-$algebra which contains all balls, and as a consequence all Borel sets.

Theorem 3.8. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space. For the binary classification setting, the learning rule 2C1NN is universally consistent for all processes $\mathbb{X} \in SMV_{(\mathcal{X},\rho)}$.*

Proof Let $\mathbb{X} \in \text{SMV}_{(\mathcal{X},\rho)}$. We will show that 2C1NN is universally consistent on $\mathbb{X}$ by considering the set $\mathcal{S}_{\mathbb{X}}$ of functions for which it is consistent. More precisely, since $\mathcal{Y} = \{0, 1\}$ in the binary setting, all target functions can be described as $f^* = \mathbb{1}_{A_{f^*}}$ where $A_{f^*} = f^{<-1>}(\{1\})$. We define $\mathcal{S}_{\mathbb{X}}$ using the corresponding sets:

$$\mathcal{S}_{\mathbb{X}} := \{A \in \mathcal{B}, \quad \mathcal{L}_{\mathbb{X}}(2C1NN, \mathbb{1}_{.\in A}) = 0 \quad (a.s.)\}$$

By construction we have $\mathcal{S}_{\mathbb{X}} \subset \mathcal{B}$. The goal is to show that in fact $\mathcal{S}_{\mathbb{X}} = \mathcal{B}$. To do so, we will show that $\mathcal{S}$ satisfies the following properties

- $\emptyset \in \mathcal{S}_{\mathbb{X}}$ and $\mathcal{S}_{\mathbb{X}}$ contains all balls $B(x, r)$ with $x \in \mathcal{X}$ and $r \geq 0$,
- if $A \in \mathcal{S}_{\mathbb{X}}$ then $A^c \in \mathcal{S}_{\mathbb{X}}$ (stable to complementary),
- if $(A_i)_{i\geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$, then $\bigcup_{i\geq 1} A_i \in \mathcal{S}_{\mathbb{X}}$ (stable to $\sigma-$additivity for disjoint sets),
- if $A, B \in \mathcal{S}_{\mathbb{X}}$, then $A \cup B \in \mathcal{S}_{\mathbb{X}}$ (stable to union).

Together, these properties show that $\mathcal{S}_{\mathbb{X}}$ is a $\sigma-$algebra that contains all open intervals of $\mathcal{X}$. Recall that by definition, $\mathcal{B}$ is the smallest $\sigma-$algebra containing open intervals. Therefore we get $\mathcal{B} \subset \mathcal{S}_{\mathbb{X}}$ which proves the theorem. We now show the four properties.

The invariance to complementary and to finite union can be shown with the same proof as Theorem 3.5. Further, we clearly have $\emptyset \in \mathcal{S}_{\mathbb{X}}$. Now let $x \in \mathcal{X}$ and $r \geq 0$, Proposition 3.3 shows that $B(x, r) \in \mathcal{S}_{\mathbb{X}}$.

We now turn to the $\sigma-$additivity for disjoint sets. Let $(A_i)_{i\geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$. We denote $A := \bigcup_{i\geq 1} A_i$. We consider the target function $f^* = \mathbb{1}_A$. We write the average loss in the following way,

$$\frac{1}{T}\sum_{t=1}^{T} \ell_{01}(2C1NN(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t), f^*(X_t)) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{X_t\in A}\mathbb{1}_{X_{\phi(t)}\notin A} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A}.$$

where the first term corresponds to type 1 errors and the second term corresponds to type 2 errors.

We suppose by contradiction that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > 0) := \delta > 0$ Therefore, there exists $\epsilon > 0$ such that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > \epsilon) \geq \frac{\delta}{2}$. We denote this event by $\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}(2C1NN, f^*) > \epsilon\}$. We first analyze the errors induced by one set A_i only. We have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A} + \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i}) &\leq \frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A_i} + \mathbb{1}_{X_t\notin A_i}\mathbb{1}_{X_{\phi(t)}\in A_i}) \\ &= \frac{1}{T}\sum_{t=1}^{T}\ell_{01}(2C1NN(\mathbb{X}_{<t}, \mathbb{1}_{\mathbb{X}_{<t}\in A_i}, X_t), \mathbb{1}_{X_t\in A_i}).\end{aligned}$$

Then, because 2C1NN is consistent for $\mathbb{1}_{.\in A_i}$, we get

$$\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A} + \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i}) \to 0 \quad (a.s.).$$

We take $\epsilon_i = \frac{\epsilon}{4\cdot 2^i}$. The above equation gives T^i such that

$$\mathbb{P}\left[\bigcap_{T\geq T^i}\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A} + \mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i}) < \epsilon_i\right\}\right] \geq 1 - \frac{\delta}{8\cdot 2^i}.$$

We will denote by $\mathcal{E}_i$ this event. We now consider the scale of the process $\mathbb{X}_{\leq T^i}$ when falling in A_i, by introducing $\eta_i > 0$ such that

$$\mathbb{P}\left[\min_{\substack{t_1,t_2\leq T^i;\, X_{t_1},X_{t_2}\in A_i;\\ X_{t_1}\neq X_{t_2}}} \rho(X_{t_1}, X_{t_2}) > \eta_i\right] \geq 1 - \frac{\delta}{8\cdot 2^i}.$$

We denote by $\mathcal{F}_i$ this event. By the union bound, we have $\mathbb{P}(\bigcup_{i\geq 1}\mathcal{E}_i^c \cup \bigcup_{i\geq 1}\mathcal{F}_i^c) \leq \frac{\delta}{4}$. Therefore, we obtain $\mathbb{P}(\mathcal{A}\cap\bigcap_{i\geq 1}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i) \geq \mathbb{P}(\mathcal{A}) - \mathbb{P}(\bigcup_{i\geq 1}\mathcal{E}_i^c\cup\bigcup_{i\geq 1}\mathcal{F}_i^c) \geq \frac{\delta}{4}$. We now construct a partition $\mathcal{P}$ obtained by subdividing each set A_i according to scale η_i. Because $\mathcal{X}$ is separable, there exists a sequence of points $(x^j)_{j\geq 1}$ in $\mathcal{X}$ such that $\forall x\in\mathcal{X}, \inf_{j\geq 1}\rho(x, x^j) = 0$. We construct the following partition of $\mathcal{X}$ given by

$$\mathcal{P} \;:\quad A^c \cup \bigcup_{i\geq 1}\bigcup_{j\geq 1}\left\{\left(B\left(x^j, \frac{\eta_i}{2}\right)\cap A_i\right)\setminus\bigcup_{k<j}B\left(x^k, \frac{\eta_i}{2}\right)\right\}.$$

Let us now consider a realization of $\boldsymbol{x}$ of $\mathbb{X}$ in the event $\mathcal{A}\cap\bigcap_{i\geq 1}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i$. The sequence $\boldsymbol{x}$ is now not random anymore. Our goal is to show that $\boldsymbol{x}$ does not visit a sublinear number of sets in the partition $\mathcal{P}$.

By construction, the event $\mathcal{A}$ is satisfied, therefore there exists an increasing sequence of times $(t_k)_{k\geq 1}$ such that for any $k\geq 1$, $\frac{1}{t_k}\sum_{t=1}^{t_k}\ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \mathbb{1}_{\boldsymbol{x}_{<t}\in A}, x_t), \mathbb{1}_{x_t\in A}) > \frac{\epsilon}{2}$. Therefore, we obtain for any $k\geq 1$,

$$\sum_{i\geq 1}\frac{1}{t_k}\sum_{t=1}^{t_k}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}) > \frac{\epsilon}{2}.$$

Also, because the events $\mathcal{E}_i$ are met, we have

$$\sum_{i\geq 1;\ t_k\geq T^i} \frac{1}{t_k}\sum_{t=1}^{t_k}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}) < \sum_{i\geq 1, t_k\geq T^i} \epsilon_i \leq \frac{\epsilon}{4}.$$

Combining the two above equations gives

$$\frac{1}{t_k}\sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}) > \frac{\epsilon}{4}. \tag{3.2}$$

We now consider the set of times such that an input point fell into the set A_i with $T^i > t_k$, either creating a mistake in the prediction of 4C1NN or inducing a later mistake within time horizon t_k: $\mathcal{T} := \bigcup_{i\geq 1;\ T^i>t_k} \mathcal{T}_i$ where

$$\mathcal{T}_i := \left\{t \leq t_k,\ x_t \in A_i,\ \left(x_{\phi(t)} \notin A \text{ or } \exists t < u \leq t_k \text{ s.t. } \phi(u) = t,\ x_u \notin A\right)\right\}.$$

We now show that all points x_t for $t \in \mathcal{T}$ fall in distinct sets of the partition $\mathcal{P}$. Indeed, because the sets A_i are disjoint, it suffices to check that for any $i \geq 1$ such that $T^i > t_k$, the points x_t for $t \in \mathcal{T}_i$ fall in distinct of the following sets

$$P_{i,j} := \left(B\left(x^j, \frac{\eta_i}{2}\right) \cap A_i\right) \setminus \bigcup_{k<j} B\left(x^k, \frac{\eta_i}{2}\right), \quad j \geq 1.$$

Note that for any $t_1 < t_2 \in \mathcal{T}_i$ we have $x_{t_1}, x_{t_2} \in A_i$ and $x_{t_1} \neq x_{t_2}$. Indeed, we cannot have $x_{t_2} = x_{t_1}$ otherwise 2C1NN would make no mistake at time t_2 and x_{t_2} would induce no future mistake either (recall that if an input point was already visited, we use simple memorization for the prediction and do not add it to the dataset). Therefore, because the event $\mathcal{F}_i$ is satisfied, for any $t_1 < t_2 \in \mathcal{T}_i$ we have $\rho(x_{t_1}, x_{t_2}) > \eta_i$. Now suppose that x_{t_1}, x_{t_2} fall in the same set $P_{i,j}$ for $j \geq 1$, then we have $\rho(x_{t_1}, x_{t_2}) \leq \rho(x^i, x_{t_1}) + \rho(x^i, x_{t_2}) < \eta_i$, which is absurd. Therefore, all points $\{x_t,\ t \in \mathcal{T}\}$ lie in different sets of the partition $\mathcal{P}$. Therefore,

$$|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}|.$$

We now lower bound $|\mathcal{T}|$, which will uncover the main interest of the learning rule 2C1NN. Intuitively, any input point incurs at most $1 + 2 = 3$ mistakes, contrary to the traditional 1NN learning rule. We now formalize this intuition.

$$\begin{aligned}
&\sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A}\mathbb{1}_{x_{\phi(t)}\in A_i}) \\
&\qquad = \sum_{t=1}^{t_k}\sum_{i\geq 1;\ t_k<T^i}\left(\mathbb{1}_{x_t\in A_i}\mathbb{1}_{x_{\phi(t)}\notin A} + \sum_{t<u\leq t_k}\mathbb{1}_{x_u\notin A}\mathbb{1}_{x_t\in A_i}\mathbb{1}_{\phi(u)=t}\right) \\
&\qquad = \sum_{i\geq 1;\ T^i>t_k}\sum_{t\leq t_k,\ x_t\in A_i}\left(\mathbb{1}_{x_{\phi(t)}\notin A} + \sum_{t<u\leq t_k}\mathbb{1}_{x_u\notin A}\mathbb{1}_{\phi(u)=t}\right) \\
&\qquad \leq \sum_{i\geq 1;\ T^i>t_k}\sum_{t\leq t_k,\ x_t\in A_i} 3\max\left(\mathbb{1}_{x_{\phi(t)}\notin A}, \mathbb{1}_{x_u\notin A}\mathbb{1}_{\phi(u)=t},\ t<u\leq t_k\right) \\
&\qquad = 3|\mathcal{T}|
\end{aligned}$$

where in the last inequality we used the fact that a given time t can have at most 2 children i.e. $|\{u > t, \phi(u) = t\}| \leq 2$ with the 2C1NN learning rule. We now use Eq (3.2) to obtain

$$|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}| \geq \frac{\epsilon}{12} t_k.$$

This holds for any $k \geq 1$. Therefore, because $t_k \to \infty$ as $k \to \infty$ we get $|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq T} \neq \emptyset\}| \neq o(T)$. Finally, this holds for any realization of $\mathbb{X}$ in the event $\mathcal{A} \cap \bigcap_{i \geq 1} \mathcal{E}_i \cap \bigcap_{i \geq 1} \mathcal{F}_i$. Therefore,

$$\mathbb{P}(|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq T} \neq \emptyset\}| = o(T)) \leq \mathbb{P}\left[\left(\mathcal{A} \cap \bigcap_{i \geq 1} \mathcal{E}_i \cap \bigcap_{i \geq 1} \mathcal{F}_i\right)^c\right] \leq 1 - \frac{\delta}{4} < 1.$$

Therefore, $\mathbb{X} \notin \text{SMV}_{(\mathcal{X},\rho)}$ which contradicts the hypothesis. This concludes the proof that

$$\mathcal{L}_{\mathbb{X}}(2C1NN, \mathbb{1}_A) = 0 \quad (a.s.),$$

and hence, $\mathcal{S}_{\mathbb{X}}$ satisfies the disjoint $\sigma-$additivity property. This ends the proof of the theorem. ■

In particular, Theorem 3.8 shows that $\text{SMV}_{(\mathcal{X},\rho)} \subset \text{SOUL}_{(\mathcal{X},\rho),([0,1],\ell_{01})}$. Together with Proposition 3.1, this shows that the set of learnable processes for binary classification is exactly $\text{SMV}_{(\mathcal{X},\rho)}$. As a result, 2C1NN is optimistically universal for binary classification.

3.6.1 Missing proofs from Proposition 3.3

Lemma 3.2. *$(P_i)_{i \geq 1}$ forms a partition of $B(\bar{x}, r)$.*

Proof These sets are clearly disjoint. Now let $x \in B(\bar{x}, r)$ and consider $j \geq 0$ such that $\frac{r}{2^{j+1}} < r - \rho(x, \bar{x}) \leq \frac{r}{2^j}$. Then, let $i \geq 1$ such that

$$\rho(x^i, x) < \min\left(\mu_{j+1}, r - \frac{r}{2^{j+1}} - \rho(x, \bar{x}), \rho(x, \bar{x}) - r + \frac{r}{2^{j-1}}\right).$$

We have $\rho(x^i, \bar{x}) \leq \rho(x^i, x) + \rho(x, \bar{x}) < r - \frac{r}{2^{j+1}}$, hence $r - \frac{r}{2^{l_i}} < r - \frac{r}{2^{j+1}}$ i.e. $l_i \leq j$. Then, we obtain $\rho(x^i, x) < \mu_{j+1} \leq \mu_{l_i+1}$ which gives $x \in B(x^i, z^i)$. Last, we observe that $\rho(x^i, \bar{x}) \geq \rho(x, \bar{x}) - \rho(x^i, \bar{x}) > r - \frac{r}{2^{j-1}}$. Therefore, $r - \frac{r}{2^{l_i+1}} > r - \frac{r}{2^{j-1}}$ i.e. $l_i + 1 \geq j$. Therefore, we have

$$\rho(x, \bar{x}) < r - \frac{r}{2^{j+1}} \leq r - \frac{r}{2^{l_i+2}},$$

which shows $x \in R_i = \bigcup_{k \leq i} P_k$. This ends the proof that $(P_i)_{i \geq 1}$ forms a partition of $B(\bar{x}, r)$. ■

Lemma 3.3. *$(A_i)_{i \geq 0}$ forms a partition of $\mathcal{X}$.*

Proof We start by proving that the sets are disjoint. By construction, if $1 \leq j < i$, we have $A_i \subset B(x^j, r^j)$, therefore $A_i \cap A_j = \emptyset$ by construction. Further, for $i \geq 1$, if $\rho(x^i, \bar{x}) > r$, we

first note that $r^i > 0$. Indeed, if $r^i = 0$, then there exists a sequence of points x_j for $j \geq 1$ such that $\rho(x_j, \bar{x}) \leq r$ and $\rho(x^i, x_j) \to 0$ as $j \to \infty$. By triangle inequality,

$$\rho(x^i, \bar{x}) \leq \rho(x^i, x_j) + \rho(x_j, \bar{x}) \leq \rho(x^i, x_j) + r.$$

This holds for any $j \geq 1$, therefore we obtain $\rho(x^i, \bar{x}) \leq r$ which contradicts our hypothesis. Therefore $r^i > 0$. Further, we have $r^i < \inf_{x:\, \rho(x,\bar{x}) \leq r} \rho(x^i, x)$. Therefore, for any $x \in A_0 = S(\bar{x}, r)$, we have $\rho(x^i, x) > r^i$ which implies $x \notin B(x^i, r^i)$. Hence, $A_0 \cap A_i = \emptyset$. Now if $\rho(x^i, \bar{x}) < r$ we show again that $r^i > 0$. Similarly, if this is not the case, we have a sequence x_j for $j \geq 1$ such that $\rho(x_j, \bar{x}) \geq r$ and $\rho(x^i, x_j) \to 0$ as $j \to \infty$. Then, observing that

$$\rho(x^i, \bar{x}) \geq \rho(x^i, x_j) - \rho(x^i, x_j) \geq r - \rho(x^i, x_j).$$

This holds for any $j \geq 1$, therefore we obtain $\rho(x^i, \bar{x}) \geq r$ which contradicts our hypothesis. This shows $r^i > 0$. Now for $x \in A_0$, we have by construction $r^i < \rho(x^i, x)$ which gives $x \notin A_i$. Hence $A_0 \cap A_i = \emptyset$. Finally, if $\rho(x^i, \bar{x}) = r$, we have $r^i = 0$ so $A_i = \emptyset$ and we obtain direly $A_0 \cap A_i = \emptyset$. This ends the proof that for any $0 \leq i < j$, we have $A_i \cap A_j = \emptyset$.

We now prove that $\cup_{i \geq 0} A_i = \mathcal{X}$. Let $x \in \mathcal{X}$. If $\rho(x, \bar{x}) = r$ then $x \in A_0$. If $\rho(x, \bar{x}) > r$ (resp. $\rho(x, \bar{x}) < r$), using the same arguments as above, we can show that we have $\inf_{\tilde{x}:\, \rho(\tilde{x},\bar{x}) \leq r} \rho(x, \tilde{x}) > 0$ (resp. $\inf_{\tilde{x}:\, \rho(\tilde{x},\bar{x}) \geq r} \rho(x, \tilde{x}) > 0$). Therefore, we let $i \geq 1$ so that we obtain $\rho(x^i, x) < \frac{1}{1+\frac{2}{c_\epsilon}} \inf_{\tilde{x}:\, \rho(\tilde{x},\bar{x}) \leq r} \rho(x, \tilde{x})$ (resp. $\rho(x^i, x) < \frac{1}{1+\frac{2}{c_\epsilon}} \inf_{\tilde{x}:\, \rho(\tilde{x},\bar{x}) \geq r} \rho(x, \tilde{x})$). This is possible because the sequence $(x^i)_{i \geq 1}$ is dense in $\mathcal{X}$. Then, we have for any $\tilde{x}$ such that $\rho(\tilde{x}, \bar{x}) \leq r$ (resp. $\rho(\tilde{x}, \bar{x}) \geq r$),

$$\rho(x^i, \tilde{x}) \geq \rho(x, \tilde{x}) - \rho(x^i, x) > \left(1 + \frac{2}{c_\epsilon} - 1\right) \rho(x^i, x) = \frac{2}{c_\epsilon} \rho(x^i, x).$$

Therefore, $r^i \geq 2\rho(x^i, x) > \rho(x^i, x)$ which gives $x \in B(x^i, r^i)$. Now note that $\bigcup_{1 \leq j \leq i} A_i = \bigcup_{1 \leq j \leq i} B(x^i, r^i)$, therefore we obtain $x \in \bigcup_{1 \leq j \leq i} A_i$. This ends the proof that $(A_i)_{i \geq 0}$ forms a partition of $\mathcal{X}$. ■

3.7 Reduction from General Value Spaces to the Binary Classification Case

In the previous sections, we showed that 2C1NN is optimistically universal and SOUL = SMV for binary classification. Our goal here is to show that the choice of binary classification is in fact not restrictive and we aim to show that the set SOUL of input processes $\mathbb{X}$ admitting universal learning is invariant to the choice of value space subject to the loss being bounded. Specifically, to show that $\text{SOUL}_{(\mathcal{Y},\ell)} \subset \text{SOUL}_{(\mathcal{Y}',\ell')}$, one aims to construct a universally consistent learning rule for $(\mathcal{Y}', \ell')$ from a universally consistent learning rule for $(\mathcal{Y}, \ell)$ under any fixed process $\mathbb{X} \in \text{SOUL}_{(\mathcal{Y},\ell)}$.

3.7.1 Prior reductions to classification settings

We first recall two important known inclusions that hold for any bounded loss setup $(\mathcal{Y}, \ell)$. The first result compares the general setting to binary classification.

Proposition 3.4 ([Han21a]). *For any separable near-metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$,*

$$\textsc{Soul}_{(\mathcal{Y},\ell)} \subset \textsc{Soul}_{(\{0,1\},\ell_{01})}.$$

This shows that binary classification is in essence the easiest setting: whenever universal online learning is achievable for some setting $(\mathcal{Y}, \ell)$, the learning rule that works on this setting should be able to perform binary classification (note that we simply require $\mathcal{Y}$ to contain at least two elements). We give a formal proof for the sake of completeness, and we note that it does not require the boundedness of ℓ.

Proof Let $y^0, y^1 \in \mathcal{Y}$ such that $\ell(y^0, y^1) := \delta > 0$. It suffices to observe that measurable functions $\mathcal{X} \to \{0,1\}$ can be mapped to the measurable functions $\mathcal{X} \to \{y^0, y^1\}$ by composing with the simple mapping ϕ such that $\phi(i) = y^i$ for $i \in \{0,1\}$. Consider a sequence $\mathbb{X} \in \textsc{Soul}_{(\mathcal{Y},\ell)}$ and let $f.$ be a universal learner for $\mathbb{X}$, we will show that $\mathbb{X} \in \textsc{Soul}_{(\{0,1\},\ell_{01})}$ by using this learner to perform binary classification. We define the learning rule $\hat{f}. = (\hat{f}_t)_{t\geq 1}$ as follows, for any $x_{\leq t} \in \mathcal{X}^t$ and $y_{<t} \in \{0,1\}^{t-1}$,

$$\hat{f}_t(x_{<t}, y_{<t}, x_t) := \begin{cases} 0 & \text{if } \ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), y^0) \leq \ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), y^1) \\ 1 & \text{otherwise.} \end{cases}$$

where we used the notation $\phi(y) := (\phi(y_t))_{t\geq 1}$. Note that by relaxed triangle inequality,

$$\begin{aligned} \ell_{01}(\hat{f}_t(x_{<t}, y_{<t}, x_t), y_t) &\leq \mathbb{1}[\ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), \phi(y_t)) \geq \ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), \phi(1-y_t))] \\ &\leq \mathbb{1}[\ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), \phi(y_t)) \geq \frac{c_\ell}{2}\delta] \\ &\leq \frac{2}{c_\ell \delta}\ell(f_t(x_{<t}, \phi(y)_{<t}, x_t), \phi(y_t)). \end{aligned}$$

Then, for any measurable function $f^* : \mathcal{X} \to \{0,1\}$ we have

$$\mathcal{L}_{\mathbb{X}}^{(\{0,1\},\ell_{01})}(\hat{f}., f^*) \leq \frac{2}{c_\ell \delta}\mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f., \phi \circ f^*),$$

which by universal consistency of $f.$ shows that $\mathcal{L}_{\mathbb{X}}^{(\{0,1\},\ell_{01})}(\hat{f}., f^*) = 0$ almost surely. Hence, $\hat{f}.$ is a universal learner for the process $\mathbb{X}$ for the setting $(\{0,1\}, \ell_{01})$ i.e. $\mathbb{X} \in \textsc{Soul}_{(\{0,1\},\ell_{01})}$. ■

In the same spirit, we now recall that any process $\mathbb{X}$ admitting strong universal online learning for countable classification $(\mathbb{N}, \ell_{01})$ admits strong universal online learning on any separable value space $(\mathcal{Y}, \ell)$. Hence, countable classification is in essence the hardest setting.

Theorem 3.9 ([Han21a]). *For any separable near-metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$,*

$$\textsc{Soul}_{(\mathbb{N},\ell_{01})} \subset \textsc{Soul}_{(\mathcal{Y},\ell)}.$$

A proof of this theorem is given in ([Han21a] Theorem 45). It uses a number of intermediary lemmas that are not introduced in this chapter. Instead, we provide novel arguments that greatly simplify the proof and that will have practical use in Section 3.4.

Proof We fix a process $\mathbb{X} \in \text{SOUL}_{(\mathbb{N},\ell_{0,1})}$, and let $f_\cdot^{\mathbb{N}}$ be the corresponding strongly consistent learning rule. By separability, there exists a dense countable sequence $(y^i)_{i\geq 1}$ of $\mathcal{Y}$ i.e. such that $\forall y \in \mathcal{Y} : \inf_{i\in\mathbb{N}} \ell(y^i, y) = 0$. Following [Han21a], given a prediction task on $(\mathcal{Y}, \ell)$ and $\epsilon > 0$, we reduce it to a countable classification using the function $h_\epsilon : y \in \mathcal{Y} \mapsto \inf\{i \in \mathbb{N} : \ell(y^i, y) < \epsilon\} \in \mathbb{N}$. This allows to define the ϵ-learning rule $f_\cdot^\epsilon$ as follows: given $x_{\leq t} \in \mathcal{X}^t$ and $y_{<t} \in \mathcal{Y}^{t-1}$,

$$f_t^\epsilon(x_{\leq t}, y_{<t}, x_t) = y^{f_t^{\mathbb{N}}(x_{\leq t}, h_\epsilon(y_{<t}), x_t)}.$$

By construction, at each step if the prediction on h_ϵ is successful, the loss of f_t^ϵ is at most ϵ. If the prediction of h_ϵ fails, we can upper bound the loss by $\bar{\ell}$:

$$\ell(f_t^\epsilon(x_{\leq t}, y_{<t}, x_t), y_t) \leq \epsilon + \bar{\ell} \cdot \ell_{01}(f_t^{\mathbb{N}}(x_{\leq t}, h_\epsilon(y)_{<t}, x_t), h_\epsilon(y)_t)$$

where $h_\epsilon(y) := (h_\epsilon(y_t))_{t\geq 1}$. Therefore, for any target measurable function $f^* : \mathcal{X} \to \mathcal{Y}$, we obtain $\mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f_\cdot^\epsilon, f^*; T) \leq \epsilon + \bar{\ell}\mathcal{L}_{\mathbb{X}}^{(\mathbb{N},\ell_{01})}(f_\cdot^\epsilon, h_\epsilon \circ f^*; T)$, where $\mathcal{L}_{\mathbb{X}}^{(\mathbb{N},\ell_{01})}(f_\cdot^\epsilon, h_\epsilon \circ f^*; T) \to 0 \ \ (a.s.)$. Unfortunately, using the learning rule $f_\cdot^\epsilon$ only ensures $\mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f_\cdot^\epsilon, f^*) \leq \epsilon$ almost surely. Thus, the final learning rule will use the learning rules $f_\cdot^{\epsilon_k}$ for a sequence of ϵ_k decreasing to 0 e.g. $\epsilon_k = 2^{-k}$. Intuitively, each learning rule $f_\cdot^{\epsilon_k}$ with prediction y^i effectively predicts that the output y_t belongs to the set $\mathcal{B}_i^{\epsilon_k} := B_\ell(y^i, \epsilon_k) \setminus \bigcup_{1\leq j<i} B_\ell(y^j, \epsilon_k)$ where we used the notation $B_\ell(y, \epsilon) = \{y' \in \mathcal{Y}, \ell(y, y') < \epsilon\}$ for the "ball" induced by the loss ℓ. We now consider the learning rule on $(\mathcal{Y}, \ell)$ denoted $\hat{f}_\cdot^{(\mathcal{Y},\ell)}$ which successively checks consistency of these set predictions $f_\cdot^{\epsilon_1}, f_\cdot^{\epsilon_2}$ etc. and outputs a point $\hat{y} \in \mathcal{Y}$ close to the consistent intersection of these sets. Formally,

$$\hat{f}_t^{(\mathcal{Y},\ell)}(x_{\leq t}, y_{<t}, x_t) = f_t^{\epsilon_{\hat{p}}}(x_{\leq t}, y_{<t}, x_t) \text{ for } \hat{p} = \max\left\{1 \leq p \leq t, \bigcap_{1\leq k\leq p} \mathcal{B}^{\epsilon_k}_{f_t^{\epsilon_k}(x_{\leq t}, y_{<t}, x_t)} \neq \emptyset\right\}.$$

In this definition, the upper bound $\hat{p} \leq t$ is put for simplicity only to ensure that there is a finite maximum. We can now show that this learning rule is universally consistent.

Let $k \geq 1$. Note that if the predictions at step $t \geq k$ of $f_t^{\epsilon_l}$ were correct for all $1 \leq l \leq k$, then the true output y_t belongs to each set prediction $y_t \in \bigcap_{1\leq l\leq k} \mathcal{B}^{\epsilon_l}_{f_t^{\epsilon_l}(x_{\leq t}, y_{<t}, x_t)}$, thus $\hat{p} \geq k$. Now let any $\bar{y} \in \bigcap_{1\leq l\leq \hat{p}} \mathcal{B}^{\epsilon_l}_{f_t^{\epsilon_l}(x_{\leq t}, y_{<t}, x_t)}$, by relaxed triangle inequality we would have

$$\ell(\hat{f}_t^{(\mathcal{Y},\ell)}(x_{\leq t}, y_{<t}, x_t), y_t) \leq c_\ell(\ell(\hat{f}_t^{(\mathcal{Y},\ell)}(x_{\leq t}, y_{<t}, x_t), \bar{y}) + \ell(y_t, \bar{y})) \leq c_\ell(\epsilon_{\hat{p}} + \epsilon_k) \leq 2c_\ell\epsilon_k.$$

Hence,

$$\ell(\hat{f}_t^{(\mathcal{Y},\ell)}(x_{\leq t}, y_{<t}, x_t), y_t) \leq 2c_\ell\epsilon_k + \bar{\ell} \cdot \sum_{l=1}^{k} \ell_{01}(f_t^{\mathbb{N}}(x_{\leq t}, h_{\epsilon_k}(y)_{<t}, x_t), h_{\epsilon_k}(y)_t),$$

and for any measurable function $f^* : \mathcal{X} \to \mathcal{Y}$, we have $\mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f_\cdot^{(\mathcal{Y},\ell)}, f^*) \leq 2c_\ell\epsilon_k \ (a.s.)$. By union bound, almost surely this holds for any $k \geq 1$ simultaneously. Therefore, almost surely $\mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f_\cdot^{(\mathcal{Y},\ell)}, f^*) = 0$ and the learning rule $f_\cdot^{(\mathcal{Y},\ell)}$ is universally consistent. ■ The results of [Han21a] offer more details that are not required in the rest of the chapter but can be found in Section 3.7.2.

For any near-metric space $(\mathcal{Y}, \ell)$, the inclusions $\textsc{Soul}_{(\mathbb{N},\ell_{01})} \subset \textsc{Soul}_{(\mathcal{Y},\ell)} \subset \textsc{Soul}_{(\{0,1\},\ell_{01})}$ given in Proposition 3.4 and Theorem 3.9 do not answer whether $\textsc{Soul}_{(\mathcal{Y},\ell_{01})}$ is invariant to the setup when the loss is bounded. The remaining question is whether $\textsc{Soul}_{(\{0,1\},\ell_{01})} \subset \textsc{Soul}_{(\mathbb{N},\ell_{01})}$ holds or not. We answer positively to this question in the next section, thereby providing a solution to the following open question.

Question 3.4 ([Han21a]). *Is the set* Soul *invariant to the specification of* $(\mathcal{Y}, \ell)$*, subject to* $(\mathcal{Y}, \ell)$ *being separable with* $0 < \bar{\ell} < \infty$*?*

Remarks on Question 3.4. In words, the open question asks whether any universal learning task is achievable whenever universal binary classification is possible. In order to answer affirmatively it would suffice to show that the countable classification setting can be reduced to the binary classification setting. Given a process $\mathbb{X}$ admitting universal learning for binary classification and a countable classification task $f^* : \mathcal{X} \to \mathbb{N}$, a natural idea would be to solve separately each of the binary classification tasks $f^{*,i}(\cdot) = \mathbb{1}(f^*(\cdot) = i)$ for $i \in \mathbb{N}$ and to merge the results together. This proof technique works when f^* takes only a finite number of values, giving rise to the following lemma.

Lemma 3.4. *For any* $k \geq 2$, $\textsc{Soul}_{([k],\ell_{01})} = \textsc{Soul}_{(\{0,1\},\ell_{01})}$.

Proof By Proposition 3.4, it suffices to prove that any process $\mathbb{X} \in \textsc{Soul}_{(\{0,1\},\ell_{01})}$ admits universal learning in the setup $([k], \ell_{01})$. To learn an unknown function $f^* : \mathcal{X} \to [k]$, it suffices to learn the k individual binary functions which predict each class: $f^{*,i}(\cdot) := \mathbb{1}(f^*(\cdot) = i)$ where $i \in [k]$. Given a universal learner $f_\cdot$ for $\mathbb{X}$ for binary classification, We can therefore consider a universal learner for $k-$multiclass classification $\hat{f}_\cdot$ which follows the prediction of $f_\cdot$ for all functions f^i as follows: for any $x_{\leq t} \in \mathcal{X}^t$ and $y_{<t} \in [k]^{t-1}$ we pose $\hat{f}_t(x_{<t}, y_{<t}, x_t) := \arg\max_{1 \leq i \leq k} f_t(x_{<t}, \mathbb{1}(y = i)_{<t}, x_t)$ where $\mathbb{1}(y = i)_{<t}$ denotes the sequence $\mathbb{1}(y_{t'} = i)_{t' \leq t}$. We can note that this learner makes a mistake only if $f_\cdot$ made a mistake in the prediction of at least one of the functions $f^{*,i}$ for $1 \leq i \leq k$. Thus,

$$\mathcal{L}_{\mathbb{X}}^{([k],\ell_{01})}(\hat{f}_\cdot, f^*) \leq \sum_{i=1}^{k} \mathcal{L}_{\mathbb{X}}^{(\{0,1\},\ell_{01})}(f_\cdot, f^{*,i}).$$

Then, $\mathcal{L}_{\mathbb{X}}^{([k],\ell_{01})}(\hat{f}_\cdot, f^*) = 0$ almost surely by universal consistence of $f_\cdot$ which shows that $\hat{f}_\cdot$ is optimistically universal for $\mathbb{X}$ and $k-$multiclass classification. ■

Unfortunately, the proof technique used to show that finitely-many classification reduces to binary classification does not extend to countably-many classification. Indeed, the rate of convergence of the average loss on the tasks $f^{*,i}(\cdot) = \mathbb{1}(f^*(\cdot) = i)$ is not uniform across $i \in \mathbb{N}$. Thus, although we can wait for the convergence of a fixed number of these predictors—say the predictions for functions $f^{*,1}, \ldots, f^{*,k}$—we do not have any guarantee on the average losses of the predictions for the next functions $f^{*,i}$ for $i > k$. Essentially, we can only guarantee low average loss for a finite number of predictors which use binary classification.

Our proof differs substantially from this approach by considering instead a very large set of predictors—uncountably many. However, we introduce a probability distribution on these predictors, which allows to have guarantees on the average loss for the predictor with

high probability on both the stochastic process $\mathbb{X}$ and the predictor. More precisely, instead of learning the individual label i, $f^{*,i}(\cdot) = \mathbb{1}(f^*(\cdot) = i)$, we use predictors of sets of labels $\sigma \in \mathcal{P}(\mathbb{N})$ as follows: $f^*_\sigma(\cdot) = \mathbb{1}(f^*(\cdot) \in \sigma)$. We can now introduce a uniform distribution for the variable σ and test the hypothesis $f^*(x_t) = i$ by analysing the probability (in σ) of the prediction for f^*_σ to be consistent with this hypothesis i.e. $f^*_\sigma(x_t) = 1$ if $f^*(x_t) \in \sigma$ and $f^*_\sigma(x_t) = 0$ if $f^*(x_t) \notin \sigma$. Intuitively, for the right hypothesis $i^* = y_t$, this probability will be close to 1, while for a wrong hypothesis $i^* \neq y_t$ consistency either results from errors in the predictors, or that both $i, i^* \in \sigma$ or both $i, i^* \notin \sigma$ which happens with probability 1/2. This discrepancy in probability will allow to discriminate which is the true hypothesis with sublinear number of mistakes.

3.7.2 Additional background on prior reductions

In the core of the chapter, we presented the two inclusions $\textsc{Soul}_{(\mathbb{N},\ell_{01})} \subset \textsc{Soul}_{(\mathcal{Y},\ell)} \subset \textsc{Soul}_{(\{0,1\},\ell_{01})}$ shown in [Han21a] for general bounded loss settings $(\mathcal{Y}, \ell)$ (Proposition 3.4 and Theorem 3.9). The results of [Han21a] offer more details which are not useful for this chapter but give perspective on previous state of the art as well as useful intuitions. Specifically, the set $\textsc{Soul}_{(\mathcal{Y},\ell)}$ only depends on whether the value space $(\mathcal{Y}, \ell)$ is *totally bounded*. We say that $(\mathcal{Y}, \ell)$ is totally bounded if it can be covered by a finite number of $\epsilon-$balls, i.e. $\forall \epsilon > 0, \exists \mathcal{Y}_\epsilon \subset \mathcal{Y}$ s.t. $\#\mathcal{Y}_\epsilon < \infty$ and $\sup_{y\in\mathcal{Y}} \inf_{y\in\mathcal{Y}_\epsilon} \ell(y_\epsilon, y) \leq \epsilon$. Note that $(\{0,1\}, \ell_{01})$ is totally bounded whereas $(\mathbb{N}, \ell_{01})$ is not. [Han21a] proved that any setup could be reduced to these two cases.

Theorem 3.10 ([Han21a])**.** *For any separable near-metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$,*

- *If $\mathcal{Y}$ is totally bounded,* $\textsc{Soul}_{(\mathcal{Y},\ell)} = \textsc{Soul}_{(\{0,1\},\ell_{01})}$,
- *If $\mathcal{Y}$ is not totally bounded,* $\textsc{Soul}_{(\mathcal{Y},\ell)} = \textsc{Soul}_{(\mathbb{N},\ell_{01})}$.

We will now give some intuition on the first point, which reduces the totally bounded setting to $k-$multiclass classification for $k \geq 2$. Finite multiclass classification can then be reduced to binary classification through Lemma 3.4. It will be useful to keep in mind the proof technique of this reduction for our main result, though it will reveal insufficient to reduce $(\mathbb{N}, \ell_{01})$ to binary classification.

Sketch of proof of Theorem 3.10. By Theorem 3.9, we know that for any general setting, $(\mathcal{Y}, \ell)$ we have $\textsc{Soul}_{(\mathbb{N},\ell_{01})} \subset \textsc{Soul}_{(\mathcal{Y},\ell)}$. The question is now, in which cases can we further reduce the setting to binary classification? Assume that in the construction of the proof of Theorem 3.9, the partition $(\mathcal{B}^\epsilon_i)_{i\geq 1}$ of $\mathcal{Y}$ into balls of size at most $\epsilon > 0$ can always be made finite. Then, we are able to construct an universally consistent learning rule from universally consistent rules for finitely-many classification, which is equivalent to universal consistence for binary classification by Lemma 3.4. Thus, we obtain the alternative $\textsc{Soul}_{(\mathcal{Y},\ell)} = \textsc{Soul}_{(\{0,1\},\ell_{01})}$.

If this is not the case, there exists $\epsilon > 0$ and an infinite—countable— sequence $(y^k)_{k\geq 1}$ in $\mathcal{Y}$ which is $\epsilon-$separated i.e. such that $\ell(y^i, y^j) \geq \epsilon$ for any $i \neq j$. Using the mapping $\phi : \mathbb{N} \to \mathcal{Y}$ defined by $\phi(i) = y^k$ for all $k \geq 1$ similarly to the construction in the proof

of Proposition 3.4, from a universal learner $f_.$ for $(\mathcal{Y}, \ell)$ we construct a learning rule $\hat{f}_.$ for $(\mathbb{N}, \ell_{01})$, such that for any measurable function $f^* : \mathcal{X} \to \mathbb{N}$,

$$\mathcal{L}_{\mathbb{X}}^{(\mathbb{N},\ell_{01})}(\hat{f}_., f^*) \leq \frac{2}{c_\ell \epsilon} \mathcal{L}_{\mathbb{X}}^{(\mathcal{Y},\ell)}(f_., \phi \circ f^*),$$

which shows that almost surely, $\mathcal{L}_{\mathbb{X}}^{(\mathbb{N},\ell_{01})}(\hat{f}_., f^*) = 0$. Therefore, any sequence which admits universal learning for $(\mathcal{Y}, \ell)$ must admit universal learning for $(\mathbb{N}, \ell_{01})$ i.e. $\textsc{Soul}_{(\mathcal{Y},\ell)} \subset \textsc{Soul}_{(\mathbb{N},\ell_{01})}$. This ends the alternative of the theorem.

3.7.3 Final reduction from countably-infinite classification to binary classification

The last step of the core reduction is to show that countably-infinite classification is no harder than binary classification.

Theorem 3.11. $\textsc{Soul}_{(\{0,1\},\ell_{01})} \subset \textsc{Soul}_{(\mathbb{N},\ell_{01})}$.

Proof Suppose you have a process $\mathbb{X} \in \textsc{Soul}_{\{0,1\}}$. We want to show that there exists some universal learner for the input process $\mathbb{X}$ and the setting $(\mathbb{N}, \ell_{01})$. Denote by $f_. := \{f_t\}_{t=1}^{\infty}$ the universal learner in the binary classification setting $(\{0,1\}, \ell_{01})$ for sequence $\mathbb{X}$ and by $f^* : \mathcal{X} \to \mathbb{N}$ the unknown function to learn. For some subsets of outputs $S \subset \mathbb{N}$ we will consider learning the binary valued function $f_S^*(\cdot) = \mathbb{1}(f^*(\cdot) \in S)$.

Specifically, we introduce a random set $\sigma \subset \mathbb{N}$ defined on the product topology of independent Bernoullis. Let $(B_j)_{j \geq 0}$ a sequence of i.i.d. Bernoulli $\mathcal{B}(1/2)$, we define $\sigma = \{j \geq 1 : B_j = 1\}$. Based on learning the functions f_σ^* we now define a statistical test which we will use to define a learning rule for the countable classification. Precisely, given a time $t \geq 0$, define for all $i \in \mathbb{N}$,

$$p^t(x_{<t}, y_{<t}, x_t; i) := \frac{\mathbb{P}_\sigma\left[f_t(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 1 \mid i \in \sigma\right] + \mathbb{P}_\sigma\left[f_t(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 0 \mid i \notin \sigma\right]}{2}.$$

where we slightly abuse notations and write $\mathbb{1}(y \in \sigma)$ to denote $(\mathbb{1}(y_t \in \sigma))_{t \geq 1}$. Intuitively, $p^t(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t; i)$ gives the proportion of subsets σ for which the hypothesis $f^*(X_t) = i$ would be consistent with the prediction on the model trained to predict $f_\sigma^*(X_t)$. We first note that although the definition of $p^t(x_{<t}, y_{<t}, x_t; i)$ involves computing expectations over the product measure for σ, its computation can be made practical by considering the values of B_j for observed values j, i.e. $j \in \{y_{t'} : t' < t\} := \mathcal{Y}$. Indeed, we can conveniently write $p^t(x_{<t}, y_{<t}, x_t; i)$ as

$$p^t(x_{<t}, y_{<t}, x_t; i) = \frac{1}{2^{|\mathcal{Y}|}} \sum_{(b_j)_{j \in \mathcal{Y}} \in \{0,1\}^{|\mathcal{Y}|}} \mathbb{P}[f_t(x_{<t}, (b_{y_{t'}})_{t'<t}, x_t) = 1]\mathbb{1}(b_i = 1) + \mathbb{P}[f_t(x_{<t}, (b_{y_{t'}})_{t'<t}, x_t) = 0]\mathbb{1}(b_i = 0),$$

where the probability is taken on the possible randomness of the learning rule only. As a result, the function $p^t(\cdot,\cdot,\cdot;\cdot)$ can be practically computed and is also measurable.

Note that if the learning rule $f.$ had no errors we would have a simple discrimination as follows

$$\frac{\mathbb{P}_\sigma\left[f^*_\sigma(X_t)=1 \mid i\in\sigma\right]+\mathbb{P}_\sigma\left[f^*_\sigma(X_t)=0 \mid i\notin\sigma\right]}{2}=\begin{cases}1 & \text{if } f^*(X_t)=i,\\ 1/2 & \text{otherwise.}\end{cases}$$

We are now ready to define a learning rule $\hat{f}:=\{\hat{f}_t\}_{t=1}^\infty$ for countable classification as follows

$$\hat{f}_t(x_{<t},y_{<t},x_t):=\begin{cases}\min\limits_{i\in\mathbb{N}}\left\{i:\ p^t(x_{<t},y_{<t},x_t;i)>\dfrac{3}{4}\right\} & \text{if } \exists i\in\mathbb{N},\ p^t(x_{<t},y_{<t},x_t;i)>\frac{3}{4}\\ 0 & \text{otherwise.}\end{cases}$$

This is a valid measurable learning rule as a result of the measurability of $p^t(\cdot,\cdot,\cdot;\cdot)$ for all $t\geq 1$. We now show that the learning rule $\hat{f}$ is universally consistent. By hypothesis of binary classification universal consistency, for any subset $S\in\mathcal{P}(\mathbb{N})$, we have $\mathbb{P}_{\mathbb{X}}[\mathcal{L}_{\mathbb{X}}(f.,f^*_S;T)\underset{T}{\to}0]=1$. Because this result is true for any subset S, we get

$$\mathbb{P}_{\mathbb{X},\sigma}\left[\mathcal{L}_{\mathbb{X}}(f.,f^*_\sigma;T)\xrightarrow[T\to\infty]{}0\right]=1$$

where the randomness is taken on both $\mathbb{X}$ and σ – and potentially the learning process $f.$. Therefore, we have

$$\mathbb{P}_\sigma\left[\mathcal{L}_{\mathbb{X}}(f.,f^*_\sigma;T)\xrightarrow[T\to\infty]{}0\right]=1,\quad \text{a.s. in } \mathbb{X}$$

Denote by $\mathcal{E}$ this event of probability 1. We will show that on this event, the learning rule is consistent. We now fix an input trajectory $\mathbb{X}$ falling in $\mathcal{E}$ which we denote by $x=(x_t)_{t=0}^\infty$ to make clear that there is no randomness on the trajectory anymore – one can think of a deterministic process. We additionally denote $y=(y_t)_{t=0}^\infty:=(f^*(x_t))_{t=0}^\infty$ for simplicity.

By construction, for any $\epsilon>0$ we have

$$\mathbb{P}_\sigma\left[\mathcal{L}_x(f.,f^*_\sigma;t)\leq\epsilon,\quad\forall t\geq T\right]\xrightarrow[T\to\infty]{}1$$

We can then define for any ϵ a time $T_\epsilon\geq 0$ such that

$$\mathbb{P}_\sigma\left[\mathcal{L}_x(f.,f^*_\sigma;t)\leq\epsilon,\quad\forall t\geq T_\epsilon\right]\geq\frac{7}{8}.$$

We define the event $\mathcal{A}_\epsilon=\{\mathcal{L}_x(f.,f^*_\sigma;t)\leq\epsilon,\quad\forall t\geq T_\epsilon\}$. An important remark is that both T_ϵ and the event $\mathcal{A}_\epsilon$ are dependent on the specific trajectory x: the learning rate of our rule depends on the realization of the input trajectory. We will show that from time T_ϵ, the error rate of $\hat{f}$ is at most 8ϵ. Let $t\geq 0$ and $i^*_t=f^*(x_t)$ be the true (random) value that we want to predict. We have for the true value i^*_t,

$$\begin{aligned}
&p^t(x_{<t},y_{<t},x_t;i^*_t)\\
&=1-\frac{\mathbb{P}_\sigma\left[f_t(x_{<t},f^*_\sigma(x_{<t}),x_t)=0 \mid i^*_t\in\sigma\right]+\mathbb{P}_\sigma\left[f_t(x_{<t},f^*_\sigma(x_{<t}),x_t)=1 \mid i^*_t\notin\sigma\right]}{2}\\
&\geq 1-\mathbb{P}_\sigma[\bar{\mathcal{A}}_\epsilon]-\mathbb{E}_\sigma\left[\left(\mathbb{1}_{f_t(x_{<t},f^*_\sigma(x_{<t}),x_t)=0,i^*_t\in\sigma}+\mathbb{1}_{f_t(x_{<t},f^*_\sigma(x_{<t}),x_t)=1,i^*_t\notin\sigma}\right)\mathbb{1}_{\mathcal{A}_\epsilon}\right]\\
&\geq 1-\frac{1}{8}-\mathbb{E}_\sigma\left[\ell(f_t(x_{<t},f^*_\sigma(x_{<t}),x_t),f^*_\sigma(x_t))\mathbb{1}_{\mathcal{A}_\epsilon}\right]
\end{aligned}$$

However, for any $i \neq i_t^*$,

$$\begin{aligned}
&p^t(x_{<t}, y_{<t}, x_t; i) \\
&= \frac{\mathbb{P}_\sigma\left[f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t) = 1 \mid i \in \sigma\right] + \mathbb{P}_\sigma\left[f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t) = 0 \mid i \notin \sigma\right]}{2} \\
&\leq \frac{1}{2} + \mathbb{E}_\sigma\left[\mathbb{1}_{f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t)=1, i\in\sigma, i_t^*\notin\sigma} + \mathbb{1}_{f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t)=0, i\notin\sigma, i_t^*\in\sigma}\right] \\
&\leq \frac{1}{2} + \mathbb{P}_\sigma[\bar{\mathcal{A}}_\epsilon] + \mathbb{E}_\sigma\left[(\mathbb{1}_{f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t)=1, i\in\sigma, i_t^*\notin\sigma} + \mathbb{1}_{f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t)=0, i\notin\sigma, i_t^*\in\sigma})\mathbb{1}_{\mathcal{A}_\epsilon}\right] \\
&\leq \frac{1}{2} + \frac{1}{8} + \mathbb{E}_\sigma\left[\ell(f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t), f_\sigma^*(x_t))\mathbb{1}_{\mathcal{A}_\epsilon}\right]
\end{aligned}$$

Note that the term $e_t := \mathbb{E}_\sigma\left[\ell(f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t), f_\sigma^*(x_t))\mathbb{1}_{\mathcal{A}_\epsilon}\right]$ is a simple scalar. Therefore, by the previous estimates on p^t, whenever $e_t < \frac{1}{8}$, the learning rule classifies the new input point correctly: $\mathbb{1}_{\hat{f}_t(x_{<t}, y_{<t}, x_t) \neq i_t^*} \leq \mathbb{1}_{e_t \geq \frac{1}{8}}$. We will now show that the bad event $e_t \geq \frac{1}{8}$ only happens with sublinear rate in t. By construction, in $\mathcal{A}_\epsilon$, for any $t \geq T_\epsilon$,

$$\frac{1}{t}\sum_{u=1}^{t} \ell(f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t), f_\sigma^*(x_t)) \leq \epsilon.$$

Therefore, for any $t \geq T_\epsilon$, we have

$$\frac{1}{t}\sum_{u=1}^{T} e_u = \frac{1}{t}\sum_{u=1}^{t} \mathbb{E}_\sigma\left[\ell(f_t(x_{<t}, f_\sigma^*(x_{<t}), x_t), f_\sigma^*(x_t))\mathbb{1}_{\mathcal{A}_\epsilon}\right] \leq \epsilon.$$

The loss of our learning rule on trajectory x now satisfies for all $t \geq T_\epsilon$,

$$\mathcal{L}_x(\hat{f}_\cdot, f^*; t) = \frac{1}{t}\sum_{u=1}^{t} \mathbb{1}_{\hat{f}_u(x_{<u}, y_{<u}, x_u) \neq i_u^*} \leq \frac{1}{t}\sum_{u=1}^{t} \mathbb{1}_{e_u \geq \frac{1}{8}} \leq \frac{8}{t}\sum_{u=1}^{t} e_u \leq 8\epsilon.$$

Thus, $\mathcal{L}_x(\hat{f}_\cdot, f^*) \leq 8\epsilon$. Taking $\epsilon > 0$ arbitrarily small shows that $\mathcal{L}_x(\hat{f}_\cdot, f^*) = 0$ and hence, the learning rule is consistent on trajectory x. Therefore, $\hat{f}_\cdot$ is consistent on the event $\mathcal{E}$ for the input sequence $\mathbb{X}$, which has probability 1. To summarize, $\mathcal{L}_\mathbb{X}(\hat{f}_\cdot, f^*) = 0$ $(a.s.)$ for any measurable function f^*, showing that $\hat{f}_\cdot$ is universally consistent and thus $\mathbb{X} \in \textsc{Soul}_\mathbb{N}$. This ends the proof of the theorem. ■

Together with Theorem 3.9, this result closes the conjecture formulated in [Han21a] by showing that the set of universally learnable sequences SOUL is invariant with respect to the setting $(\mathcal{Y}, \ell)$ when the loss is bounded.

Theorem 3.12. *For any separable near-metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$, we have* $\textsc{Soul}_{(\mathcal{X},\rho,\mathcal{Y},\ell)} = \textsc{Soul}_{(\mathcal{X},\rho,\{0,1\},\ell_{01})}$.

In particular, to characterize the set SOUL it suffices to focus on universal binary classification. Further, the reduction is constructive and in the proof of Theorem 3.9 and Proposition 3.4 all learning rules were constructed independently from the stochastic process $\mathbb{X}$. We can check that this in turn provides a construction of an optimistically universal learning rule for any setting $(\mathcal{Y}, \ell)$ given an optimistically universal learning rule for binary classification.

Theorem 3.13. *The existence of an optimistically universal learning rule is invariant to the output space $(\mathcal{Y}, \ell)$ when $0 < \bar{\ell} < \infty$. In particular, provided an optimistically universal learning rule for binary classification $(\{0,1\}, \ell_{01})$ one can construct an optimistically universal learning rule for a general setup $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$.*

Proof We start by supposing that there exists an optimistically universal learning rule $f_\cdot^{\{0,1\}}$ for the binary classification setting, and now construct an optimistically universal learning rule for a general setting $(\mathcal{Y}, \ell)$ satisfying $0 < \bar{\ell} < \infty$. This results from the fact that the construction in the proofs of both Theorem 3.11 and Theorem 3.9 are invariant to $\mathbb{X}$. Precisely, we first construct an optimistically universal learning rule for countably-many classification as given in the proof of Theorem 3.9. With

$$p^t(x_{<t}, y_{<t}, x_t; i) := \frac{1}{2}\left(\mathbb{P}_\sigma\left[f_t^{\{0,1\}}(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 1 \mid i \in \sigma\right] + \mathbb{P}_\sigma\left[f_t^{\{0,1\}}(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 0 \mid i \notin \sigma\right]\right),$$

we define

$$f_t^{\mathbb{N}}(x_{<t}, y_{<t}, x_t) := \begin{cases} \min_{i \in \mathbb{N}} \left\{ i,\ p^t(x_{<t}, y_{<t}, x_t; i) > \frac{3}{4} \right\} & \text{if } \exists i \in \mathbb{N},\ p^t(x_{<t}, y_{<t}, x_t; i) > \frac{3}{4} \\ 0 & \text{otherwise.} \end{cases}$$

By construction, and Theorem 3.11, $f_\cdot^{\mathbb{N}}$ is an optimistically universal learning rule for $(\mathbb{N}, \ell_{01})$. We now use the construction given by Theorem 3.9 to get an optimistically universal learning rule $f_\cdot^{(\mathcal{Y},\ell)}$ for $(\mathcal{Y}, \ell)$. Define a sequence $(y^i)_{i \geq 1}$ dense in $\mathcal{Y}$ with respect to ℓ. For $k \geq 1$ and $\epsilon_k = 2^{-k}$, we define the functions $h_k(y) = \inf\{i \geq 1, l(y^i, y) < \epsilon_k\}$ and construct the learning rules $f_\cdot^k$ by

$$f_t^k(x_{<t}, y_{<t}, x_t) = y^{f_t^{\mathbb{N}}(x_{<t}, h_k(y_{<t}), x_t)}.$$

Denoting by $B_\ell(y, \epsilon) = \{y' \in \mathcal{Y}, \ell(y, y') < \epsilon\}$ and $\mathcal{B}_i^k := B_\ell(y_i, \epsilon_k) \setminus \bigcup_{1 \leq j < i} B_\ell(y_i, \epsilon_k)$, we now define our final learning rule

$$f_t^{(\mathcal{Y},\ell)}(x_{\leq t}, y_{<t}, x_t) = f_t^{\hat{p}}(x_{\leq t}, y_{<t}, x_t) \text{ for } \hat{p} = \max\left\{1 \leq p \leq t, \bigcap_{1 \leq k \leq p} \mathcal{B}^k_{f_t^k(x_{\leq t}, y_{<t}, x_t)} \neq \emptyset\right\},$$

which is invariant to the process $\mathbb{X}$, hence optimistically universal by the proof of Theorem 3.9.

We now show the converse. Suppose there exists some setting $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$ admitting an optimistically universal learner $f_\cdot^{(\mathcal{Y},\ell)}$. We will construct an optimistically universal learning rule for binary classification using the proof of Proposition 3.4. Let $y^0, y^1 \in \mathcal{Y}$ such that $\ell(y^0, y^1) > 0$ and consider the function defined by $\phi(i) = y^i$ for $i \in \{0,1\}$. We now construct a learning rule $f_\cdot^{\{0,1\}}$ for binary classification as follows

$$f_t^{\{0,1\}}(x_{<t}, y_{<t}, x_t) := \begin{cases} 0 & \text{if } \ell(f_t^{(\mathcal{Y},\ell)}(x_{<t}, \phi(y)_{<t}, x_t), y^0) \leq \ell(f_t^{(\mathcal{Y},\ell)}(x_{<t}, \phi(y)_{<t}, x_t), y^1) \\ 1 & \text{otherwise.} \end{cases}$$

This learning rule is invariant to $\mathbb{X}$, hence optimistically universal by the proof of Proposition 3.4. This ends the proof of the theorem. ■

3.7.4 Learning rules preserved by the reduction

Though its definition is little abstruse, the countable classification learning rule that is derived from the proof of Theorem 3.9 leaves many learning rules unchanged. In particular, the following proposition shows that learning rules based on a representant which depends only on the historical input sequence e.g. nearest neighbor rule, are transported by our construction.

Proposition 3.5. *Let $\{f_t\}_{t=1}^{\infty}$ be a learning rule defined by representant function $\phi(t) \in \{1, ..., t-1\}$ which at step t only depends on $(x_1, .., x_t)$ as follows,*

$$f_t(x_{<t}, y_{<t}, x_t) = y_{\phi(t)}.$$

Note that this learning rule can be defined for any output setting $(\mathcal{Y}, \ell)$. If $\{f_t\}_{t=1}^{\infty}$ is universally consistent on a process $\mathbb{X}$ for binary classification, it is also universally consistent on $\mathbb{X}$ for any separable near-metric setting $(\mathcal{Y}, \ell)$ with bounded loss.

Proof We first show that the learning rule $f_\cdot = \{f_t\}_{t=1}^{\infty}$ is transported by our construction in Theorem 3.11 for classification with countable number of classes. In the rest of the proof, we will denote by $\phi(\cdot)$ the representant function of $f_\cdot$. With

$$p^t(x_{<t}, y_{<t}, x_t; i) := \frac{1}{2}\left(\mathbb{P}_\sigma\left[f_t(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 1 \mid i \in \sigma\right]\right.$$
$$\left. + \mathbb{P}_\sigma\left[f_t(x_{<t}, \mathbb{1}(y \in \sigma)_{<t}, x_t) = 0 \mid i \notin \sigma\right]\right),$$

we define our learning rule $f_\cdot^{\mathbb{N}} := \{f_t^{\mathbb{N}}\}_{t=1}^{\infty}$ for countably-many classification as in Theorem 3.11:

$$f_t^{\mathbb{N}}(x_{<t}, y_{<t}, x_t) := \begin{cases} \min\limits_{i \in \mathbb{N}} \left\{ i,\ p^t(x_{<t}, y_{<t}, x_t; i) > \dfrac{3}{4} \right\} & \text{if } \exists i \in \mathbb{N},\ p^t(x_{<t}, y_{<t}, x_t; i) > \frac{3}{4} \\ 0 & \text{otherwise.} \end{cases}$$

We now show that $f_\cdot^{\mathbb{N}}$ is in fact defined with a similar representant function. Indeed,

$$p^t(x_{<t}, y_{<t}, x_t; i) = \frac{\mathbb{P}_\sigma\left[\mathbb{1}(y_{\phi(t)} \in \sigma) = 1 \mid i \in \sigma\right] + \mathbb{P}_\sigma\left[\mathbb{1}(y_{\phi(t)} \in \sigma) = 0 \mid i \notin \sigma\right]}{2}$$
$$= \begin{cases} 1 \text{ if } y_{\phi(t)} = i \\ \frac{1}{2} \text{ if } y_{\phi(t)} \neq i \end{cases}$$

Therefore, we obtain $f_t^{\mathbb{N}}(x_{<t}, y_{<t}, x_t) = y_{\phi(t)}$, which shows that $f_\cdot^{\mathbb{N}} = f_\cdot$ i.e. that the learning rule $f_\cdot$ is transported by the construction.

We now fix a separable near-metric space $(\mathcal{Y}, \ell)$ and a process $\mathbb{X}$ such that $f_\cdot$ is universally consistent for binary classification. By the above arguments, Theorem 3.11 shows that $f_\cdot$ is also universally consistent for countable classification. We now aim to show that $f_\cdot$ on $(\mathcal{Y}, \ell)$ is universally consistent on $\mathbb{X}$. Let f^* be a measurable target function and $\epsilon > 0$. We take a sequence $(y^i)_{i \geq 1}$ dense on $\mathcal{Y}$ with respect to ℓ and construct the function $h(y) = \inf\{i \geq$

$1, \ell(y^i, y) < \epsilon\}$. Then, $y^{f_t(x_{<t}, h_k(y_{<t}), x_t)} = y^{h(y_{\phi(t)})}$. Hence, if $f_t(x_{<t}, h(y_{<t}), x_t) = h(y_t)$ we obtain $y^{h(y_{\phi(t)})} = y^{h(y_t)}$. Therefore, we can write

$$\begin{aligned}\ell(y_{\phi(t)}, y_t) &\leq \bar{\ell} \cdot \mathbb{1}_{f_t(x_{<t}, h(y_{<t}), x_t) \neq h(y_t)} + \ell(y_{\phi(t)}, y_t) \mathbb{1}_{f_t(x_{<t}, h(y_{<t}), x_t) = h(y_t)} \\ &\leq \bar{\ell} \cdot \mathbb{1}_{f_t(x_{<t}, h(y_{<t}), x_t) \neq h(y_t)} + c_\ell(\ell(y_{\phi(t)}, y^{h(y_{\phi(t)})}) + \ell(y^{h(y_t)}, y_t)) \\ &\leq \bar{\ell} \cdot \ell_{01}(f_t(x_{<t}, h(y_{<t}), x_t), h(y_t)) + 2c_\ell\epsilon.\end{aligned}$$

This yields $\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T) \leq \bar{\ell}\mathcal{L}_{\mathbb{X}}(f_\cdot, h \circ f^*; T) + 2c_\ell\epsilon$. Because $f_\cdot$ is universally consistent for the setting $(\mathbb{N}, \ell_{01})$, it is in particular consistent for target function $h \circ f^* : \mathcal{X} \to \mathbb{N}$. Therefore, $\limsup_T \mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T) \leq 2c_\ell\epsilon$, $(a.s.)$. This is valid for $\epsilon_k = 2^{-k}$ for all $k \geq 1$. Therefore, by union bound, $\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T) \to 0$, $(a.s.)$, which ends the proof that $f_\cdot$ is universally consistent on $\mathbb{X}$ for the setting $(\mathcal{Y}, \ell)$. ■

We are now ready to apply these general reduction results to the kC1NN algorithms that we constructed in Section 3.5 and showed were optimistically universal for binary classification in Section 3.6. Indeed, these learning rules exactly use representants, and can therefore be rewritten as in Proposition 3.5. Together with Theorem 3.12 we obtain a full characterization of the set of processes admitting strong universal learning for general value spaces and obtain that 2C1NN is optimistically universal for general input and output spaces.

Corollary 3.2. *For any separable Borel space $\mathcal{X}$ and any separable near-metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$, we have $\textsc{Soul}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} = \textsc{SMV}_{(\mathcal{X},\rho)}$. Further, 2C1NN is an optimistically universal learning rule.*

Note that the case $\bar{\ell} = 0$ can be treated separately: in this case all processes $\mathbb{X}$ are learnable and any learning rule is optimistically universal. The same proofs imply that the learning rules kC1NN are also optimistically universal for any $k \geq 2$. As a remark, one can note that 2C1NN is the simplest algorithm of this class which is optimistically universal. Indeed, 1C1NN systematically deletes the previous points from the dataset and as a result, at any time, the dataset $\mathcal{D}_t$ is a singleton. Hence, 1C1NN is not optimistically universal.

3.8 Weak Universal Learning

We now turn to weak universal learning. In this section, we show that the results for the characterization of learnable processes and the existence of optimistically universal learning rules for the strong setting can also be adapted to the weak setting. Although the set of learnable processes differ—$\textsc{Soul} \subset \textsc{Woul}$ in general and $\textsc{Soul} \subsetneq \textsc{Woul}$ whenever $\mathcal{X}$ is infinite [Han21a]—we show that the same learning rule 2C1NN is optimistically universal in the weak setting. We recall the necessary condition WSMV for weak learnability for near-metric separable value spaces $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$.

Condition WSMV. *For every disjoint sequence $\{A_k\}_{k=1}^\infty$ of measurable sets of $\mathcal{X}$ with $\bigcup_{k=1}^\infty A_k = \mathcal{X}$, (every countable measurable partition), $\mathbb{E}[|\{k \in \mathbb{N} : A_k \cap \mathbb{X}_{<T} \neq \emptyset\}|] = o(T)$.*

Similarly to the strong consistency case, we will show that $\text{WSMV}_{(\mathcal{X},\rho)}$ is also sufficient for weak universal consistency. Note that whenever $\mathcal{X}$ is infinite we have $\text{SMV}_{(\mathcal{X},\rho)} \subsetneq$

$\text{WSMV}_{(\mathcal{X},\rho)}$. We start by adapting Proposition 3.3 for the weak setting by showing that 2C1NN is weakly consistent on balls under any process $\mathbb{X} \in \text{WSMV}_{(\mathcal{X},\rho)}$.

Proposition 3.6. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space constructed from some metric ρ. We consider the binary classification setting $\mathcal{Y} = \{0, 1\}$ and the ℓ_{01} binary loss. For any input process $\mathbb{X} \in WSMV_{(\mathcal{X},\rho)}$, for any $x \in \mathcal{X}$, and $r > 0$, the learning rule 2C1NN is weakly consistent for the target function $f^* = \mathbb{1}_{B_\rho(x,r)}$.*

Proof The proof uses a similar structure to the proof of Proposition 3.3. We fix $\bar{x} \in \mathcal{X}$, $r > 0$ and $f^*(\cdot) = \mathbb{1}_{B(\bar{x},r)}$. We reason by the contrapositive and suppose that 2C1NN is not weakly consistent on f^*. We will show that the process $\mathbb{X}$ disproves the $\text{WSMV}_{(\mathcal{X},\rho)}$ condition.

Because 2C1NN is not weakly consistent for f^*, there exists ϵ and an increasing sequence of times $(T_l)_{l\geq 1}$ such that for any $l \geq 1$,

$$\mathbb{E}\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T_l) \geq \epsilon T_l.$$

We now define a partition $\mathcal{P}$. Because $\mathcal{X}$ is separable, there exists a sequence $(x^i)_{i\geq 1}$ of elements of $\mathcal{X}$ which is dense. We focus for now on the sphere $S(\bar{x}, r)$ and for any $\tau > 0$ we take $(P_i(\tau))_{i\geq 1}$ the sequence of sets included in $S(\bar{x}, r)$ defined by

$$P_i(\tau) := \left(S(\bar{x}, r) \cap B(x^i, \tau)\right) \setminus \left(\bigcup_{1\leq j<i} B(x^j, \tau)\right).$$

These sets form a partition of $S(\bar{x}, r)$ as shown in the proof of Proposition 3.3. We now pose $\tau_l := c_\epsilon \cdot \frac{r}{2^{l+1}}$, for $l \geq 1$, where $c_\epsilon := \frac{1}{2 \cdot 2^{2^5/\epsilon}}$ is a constant dependant on ϵ only. We also pose $\tau_0 = r$. Then, because $\mathbb{X} \in \text{WSMV}_{(\mathcal{X},\rho)}$, the expected number of sets visited of $\mathcal{P}_i(\tau_l)$ tends to 0. Therefore, there exists an increasing sequence $(n_l)_{l\geq 1}$ such that for any $l \geq 1$,

$$\forall n \geq n_l, \quad \mathbb{E}[|\{i,\ P_i(\tau_l) \cap \mathbb{X}_{<n} \neq \emptyset\}|] \leq \frac{\epsilon^2}{2^{10}} n \quad \text{and} \quad n_{l+1} \geq \frac{2^6}{\epsilon} n_l$$

Now, for any $l \geq 1$, we now construct $\mu_l > 0$ such that

$$\mathbb{P}\left[\min_{i<j\leq n_l,\ X_i \neq X_j} \rho(X_i, X_j) > \mu_l\right] \geq 1 - \frac{\epsilon}{2^{l+3}}.$$

We denote by $\mathcal{F}_l$ this event. Therefore $\mathbb{P}[\mathcal{F}_l^c] \leq \frac{\epsilon}{2^{l+3}}$. Note that the sequence $(\mu_l)_{l\geq 1}$ is non-increasing. We now define radiuses $(z^i)_{i\geq 1}$ as follows:

$$z^i = \begin{cases} \mu_{l_i+1} & \text{if } \rho(x^i, \bar{x}) < r, \text{ where } \frac{r}{2^{l_i+1}} < r - \rho(x^i, \bar{x}) \leq \frac{r}{2^{l_i}} \\ 0 & \text{if } \rho(x^i, \bar{x}) \geq r, \end{cases}$$

and consider the sets $R_i := B(x^i, z^i) \cap \left\{x \in \mathcal{X} :\ \rho(x, \bar{x}) < r - \frac{r}{2^{l_i+2}}\right\}$. We construct $P_i := R_i \setminus \left(\bigcup_{k<i} R_k\right)$, for $i \geq 1$. By Lemma 3.2, $(P_i)_{i\geq 1}$ forms a partition of $B(\bar{x}, r)$. We now

define a second partition $(A_i)_{i\geq 1}$ similarly as in the proof of Proposition 3.3. We start by defining a sequence of radiuses $(r^i)_{i\geq 1}$ as follows

$$r^i = \begin{cases} c_\epsilon \inf\limits_{x:\ \rho(x,\bar{x})\leq r} \rho(x^i, x) & \text{if } \rho(x^i,\bar{x}) > r, \\ c_\epsilon \inf\limits_{x:\ \rho(x,\bar{x})\geq r} \rho(x^i, x) & \text{if } \rho(x^i,\bar{x}) < r, \\ 0 & \text{if } \rho(x^i,\bar{x}) = r, \end{cases}$$

and consider the sets $(A_i)_{i\geq 0}$ given by $A_0 = S(\bar{x}, r)$ and for $i \geq 1$, $A_i = B(x^i, r^i) \setminus \left(\bigcup_{1\leq j<i} B(x^j, r^j)\right)$. By Lemma 3.3, this forms a partition of $\mathcal{X}$. We now formally consider the product partition of $(P_i)_{i\geq 1}$ and $(A_i)_{i\geq 0}$ i.e.

$$\mathcal{Q}: \bigcup_{i\geq 0,\ A_i\subset B(\bar{x},r)} \bigcup_{j\geq 1} (A_i \cap P_j) \cup \bigcup_{i\geq 0,\ A_i\subset \mathcal{X}\setminus B(\bar{x},r)} A_i.$$

where we used the fact that sets A_i satisfy either $A_i \subset B(\bar{x}, r)$ or $A_i \subset \mathcal{X} \setminus B(\bar{x}, r)$. We will show that this partition disproves the $\text{WSMV}_{(\mathcal{X},\rho)}$ hypothesis on $\mathbb{X}$.

We now fix $l_0 \geq 1$ such that $T_{l_0} \geq n_2$ and consider $l \geq l_0$. We focus on time T_l. Define the event $\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T_l) \geq \frac{\epsilon}{2} T_l\}$. Note that we have

$$\mathbb{E}\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*; T_l) \leq \frac{\epsilon}{2} T_l + \mathbb{P}[\mathcal{A}] T_l.$$

Therefore, $\mathbb{P}[\mathcal{A}] \geq \frac{\epsilon}{2}$. Also, because $(n_u)_{u\geq 1}$ is an increasing sequence, let $u \geq 1$ such that $n_{u+1} \leq T_l \leq n_{u+2}$. We define the event $\mathcal{E} = \{|\{i\ P_i(\tau_u) \cap \mathbb{X}_{\leq T_l} \neq \emptyset\}| \leq \frac{\epsilon}{2^7} T_l\}$. Then, we have by construction

$$\frac{\epsilon^2}{2^{10}} T_l \geq \mathbb{E}|\{i\ P_i(\tau_u) \cap \mathbb{X}_{\leq T_l} \neq \emptyset\}| \geq \frac{\epsilon}{2^7} T_l \mathbb{P}[\mathcal{E}^c].$$

Therefore, we have $\mathbb{P}[\mathcal{E}^c] \leq \frac{\epsilon}{8}$. Consider a specific realization $\boldsymbol{x} = (x_t)_{t\geq 0}$ of the process $\mathbb{X}$ falling in the event $\mathcal{A} \cap \mathcal{E} \cap \bigcap_{l\geq 1} \mathcal{F}_l$. This event has probability

$$\mathbb{P}\left[\mathcal{A} \cap \mathcal{E} \cap \bigcap_{l\geq 1} \mathcal{F}_l\right] \geq \mathbb{P}[\mathcal{A}] - \mathbb{P}[\mathcal{E}^c] - \sum_{l\geq 1} \mathbb{P}[\mathcal{F}_l^c] \geq \frac{\epsilon}{2} - \frac{\epsilon}{8} - \frac{\epsilon}{8} = \frac{\epsilon}{4}.$$

Note that $\boldsymbol{x}$ is not random anymore. We now show that $\boldsymbol{x}$ visits a large number of sets in the partition $\mathcal{Q}$. We now denote by $(t_k)_{k\geq 1}$ the increasing sequence of all times when 2C1NN makes an error in the prediction of $f^*(x_t)$. Define k_l such the last time of error before T_l i.e. $k_l = \max\{k \geq 1,\ t_k \leq T_l\}$. By construction, because $\mathcal{A}$ is met we have $k_l \geq \frac{\epsilon}{2} T_l$.

At an iteration where the new input x_t has not been previously visited we will denote by $\phi(t)$ the index of the nearest neighbor of the current dataset in the 2C1NN learning rule. Now let $l \geq 1$. Consider the tree $\mathcal{G}$ where nodes are times $\mathcal{T} := \{t,\ t \leq T_l,\ x_t \notin \{x_u, u < t\}\}$ for which a new input was visited, where the parent relations are given by $(t, \phi(t))$ for $t \in \mathcal{T}\setminus\{1\}$. Again, each node has at most 2 children and a node is not in the dataset at time T_l when it has exactly 2 children.

Step 1. We now suppose that the majority of input points on which 2C1NN made a mistake belong to the $B(\bar{x}, r)$ i.e.

$$|\{t \leq T_l,\ \ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) = 1,\ x_t \in B(\bar{x}, r)\}| \geq \frac{k_l}{2},$$

or equivalently $|\{k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}| \geq \frac{k_l}{2}$.

Let us now consider the subgraph $\tilde{\mathcal{G}}$ given by restricting $\mathcal{G}$ only to nodes in the the ball $B(\bar{x}, r)$ which are mapped to the true value 1 i.e. on times $\{t \in \mathcal{T},\ x_t \in B(\bar{x}, r)\}$. As in the proof of Proposition 3.3, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with roots times $\{t_k,\ k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}$—and possibly $t = 1$ if $x_1 \in B(\bar{x}, r)$. For a given time t_k with $k \leq k_l$ and $x_{t_k} \in B(\bar{x}, r)$, denote $\mathcal{T}_k$ the corresponding tree in $\tilde{\mathcal{G}}$ with root t_k. We will say that the tree $\mathcal{T}_k$ is *sparse* if

$$\forall t \in \mathcal{T}_k, \quad |\{u \leq T_l,\ \phi(u) = t,\ \rho(x_u, \bar{x}) < r\}| \leq 1 \quad \text{and} \quad |\mathcal{T}_k| \leq \frac{16}{\epsilon}.$$

We denote by $S = \{k \leq k_l,\ \rho(x_{t_k}, \bar{x}) < r,\ \mathcal{T}_k \text{ sparse}\}$ the set of sparse trees. Similarly as in the proof of Proposition 3.3, we have $|S| \geq \frac{k_l}{8}$. We now focus only on sparse trees $\mathcal{T}_k$ for $k \in S$ and analyze their relation with the final dataset $\mathcal{D}_{T_l+1}$. Precisely, for a sparse tree $\mathcal{T}_k$, denote $\mathcal{V}_k = \mathcal{T}_k \cap \mathcal{D}_{T_l+1}$ the set of times which are present in the final dataset and belong to the tree induced by error time t_k. Because each node of $\mathcal{T}_k$ and not present in $\mathcal{D}_{T_l+1}$ has at least 1 children in $\mathcal{T}$, we note that $\mathcal{V}_k \neq \emptyset$. We now consider the path from a node of $\mathcal{V}_k$ to the root t_k. We denote by $d(k)$ the depth of this node in $\mathcal{V}_k$ and denote the path by $p^k_{d(k)} \to p^k_{d(k)-1} \to p^k_0 = t_k$ where $p^k_{d(k)} \in \mathcal{V}_k$. Then we have, $d(k) \leq |\mathcal{T}_k| - 1 \leq \frac{16}{\epsilon} - 1$. The same arguments as in the proof of Proposition 3.3 show that all the points $\{p^k_{d(k)},\ k \in S\}$ fall in distinct sets of the partition $(A_i)_{i \geq 0}$. Therefore,

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}| \geq |S| \geq \frac{k_l}{8} \geq \frac{\epsilon}{16} T_l.$$

Step 2. We now turn to the case when the majority of input points on which 2C1NN made a mistake are not in the ball $B(\bar{x}, r)$ i.e.

$$|\{t \leq t_{k_l},\ \ell_{01}(2C1NN(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t), f^*(x_t)) = 1,\ \rho(x_t, \bar{x}) \geq r\}| \geq \frac{k_l}{2},$$

or equivalently $|\{k \leq k_l,\ \rho(x_{t_k}, \bar{x}) \geq r\}| \geq \frac{k_l}{2}$. Similarly as the previous case, we consider the graph $\tilde{G}$ given by restricting $\mathcal{G}$ only to nodes outside the ball $B(\bar{x}, r)$ i.e. on times $\{t \in \mathcal{T}, \rho(x_t, \bar{x}) \geq r\}$. Again, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with root times $\{t_k,\ k \leq k_l,\ \rho(x_{t_k}, \bar{x}) \geq r\}$—and possibly $t = 1$. We denote $\mathcal{T}_k$ the corresponding tree of $\tilde{\mathcal{G}}$ rooted in t_k. Similarly to above, a tree is *sparse* if

$$\forall t \in \mathcal{T}_k, \quad |\{u \leq T_l,\ \phi(u) = t,\ \rho(x_u, \bar{x}) < r\}| \leq 1 \quad \text{and} \quad |\mathcal{T}_k| \leq \frac{16}{\epsilon}.$$

If $S = \{k \leq k_l, ; \rho(x_{t_k}, \bar{x}) \geq r,\ \mathcal{T}_k \text{ sparse}\}$ denotes the set of sparse trees, the same proof as above shows that $|S| \geq \frac{k_l}{8}$. Again, for any $k \in S$, if $d(k)$ denotes the depth of some node

from $\mathcal{V}_k := \mathcal{T}_k \cap \mathcal{D}_{t_{k_l}}$ in $\mathcal{T}_k$ we have $d(k) \leq \frac{16}{\epsilon} - 1$. For each $k \in S$ we consider the path from this node of $\mathcal{V}_k$ to the root t_k: $p^k_{d(k)} \to p^k_{d(k)-1} \to \ldots \to p^k_0 = t_k$ where $p^k_{d(k)} \in \mathcal{V}_k$. The same proof as above shows that all the points $\{p^k_{d(k)},\ k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}$ lie in distinct sets of the partition $(A_i)_{i\geq 0}$. Suppose $|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2}$, then we have

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}| \geq |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l}{16} \geq \frac{\epsilon}{32} T_l.$$

Step 3. In this last step, we suppose again that the majority of input points on which 2C1NN made a mistake are not in the ball $B(\bar{x}, r)$ and that $|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| < \frac{|S|}{2}$. Therefore, we obtain

$$|\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}| = |S| - |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l}{16} \geq \frac{\epsilon}{32} T_l.$$

We will now make use of the partition $(P_i)_{i\geq 1}$. Recall that $u \geq 1$ was defined such that $n_{u+1} \leq T_l \leq n_{u+2}$. Note that we have $n_u \leq \frac{\epsilon}{2^6} n_{u+1} \leq \frac{\epsilon}{2^6} T_l$. Let us now analyze the process between times n_u and T_l. In particular, we are interested in the indices $T = \{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}$ and times $\mathcal{U}_u = \{p^k_{d(k)} :\ n_u < p^k_{d(k)} \leq k_l,\ k \in T\}$. We have

$$|\mathcal{U}_u| \geq |\{k \in S,\ \rho(x_{p^k_{d(k)}}, \bar{x}) = r\}| - n_u \geq \frac{\epsilon}{32} T_l - \frac{\epsilon}{2^6} T_l = \frac{\epsilon}{2^6} T_l.$$

Because the event $\mathcal{E}_u$ is met, we have

$$|\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\mathcal{U}_u} \neq \emptyset\}| \leq |\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}| \leq \frac{\epsilon}{2^7} T_l.$$

The same arguments as in the proof of Proposition 3.3 show that defining $T' := \{k \in T,\ r - \frac{r}{2^{u+2}} \leq \rho(x_{t_k}, \bar{x}) < r\}$ we obtain

$$|T'| \geq |\mathcal{U}_u| - |\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\mathcal{U}_u} \neq \emptyset\}| \geq \frac{\epsilon}{2^7} T_l.$$

We will now show that all the points in $\{x_{t_k},\ k \in T'\}$ lie in distinct sets of $(P_i)_{i\geq 1}$. Note that because we have $T_l \leq n_{u+2}$ and because the event $\mathcal{F}_{u+2}$ is met, we have that for any $p, q \in T'$ that $\rho(x_{\phi(t_p)}, x_{\phi(t_q)}) > \mu_{u+2}$. Now suppose by contradiction that $x_{\phi(t_p)}, x_{\phi(t_q)} \in P_i$ for some $i \geq 1$. Then, with l_i such that $r - \frac{r}{2^{l_i}} \leq \rho(x^i, \bar{x}) < r - \frac{r}{2^{l_i+1}}$ we have that

$$x_{\phi(t_p)}, x_{\phi(t_q)} \in \left\{x \in \mathcal{X} :\ \rho(x, \bar{x}) < r - \frac{r}{2^{l_i+2}}\right\}$$

But we know that $\rho(x_{\phi(t_p)}, \bar{x}) \geq r - \frac{r}{2^{u+2}}$. Therefore we obtain $r - \frac{r}{2^{l_i+2}} > r - \frac{r}{2^{u+2}}$ and hence $l_i \geq u+1$. Recall that $P_i \subset B(x^i, \mu_{l_i+1})$. Therefore, we obtain $\rho(x_{\phi(t_p)}, x_{\phi(t_q)}) \leq \mu_{l_i+1} \leq \mu_{u+2}$, which contradicts the fact that $\rho(x_{\phi(t_p)}, x_{\phi(t_q)}) > \mu_{u+2}$. This ends the proof that all points of $\{x_{\phi(t_k)},\ k \in T'\}$ lie in distinct subsets of $(P_i)_{i\geq 1}$. Now we obtain

$$|\{i,\ P_i \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}| \geq |T'| \geq \frac{\epsilon}{2^7} T_l.$$

Step 4. In conclusion, in all cases, we obtain

$$|\{Q \in \mathcal{Q},\ Q \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}| \geq \max(|\{i,\ A_i \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}|, |\{i,\ P_i \cap \boldsymbol{x}_{\leq T_l} \neq \emptyset\}|) \geq \frac{\epsilon}{2^7} T_l.$$

Recall that this holds for any realization $\boldsymbol{x}$ in the event $\mathcal{A} \cap \mathcal{E} \cap \bigcap_{l \geq 1} \mathcal{F}_l$. Therefore,

$$\mathbb{E}[|\{Q \in \mathcal{Q},\ Q \cap \mathbb{X}_{\leq T_l} \neq \emptyset\}|] \geq \mathbb{P}\left[\mathcal{A} \cap \mathcal{E} \cap \bigcap_{l \geq 1} \mathcal{F}_l\right] \frac{\epsilon}{2^7} T_l \geq \frac{\epsilon^2}{2^9} T_l.$$

Because this is true for all $l \geq l_0$ and T_l is an increasing sequence, we conclude that $\mathbb{X} \notin \mathrm{WSMV}_{(\mathcal{X},\rho)}$ which is absurd. Therefore 2C1NN is consistent on f^*. ■

We then show that 2C1NN is weakly consistent under processes of $\mathrm{WSMV}_{(\mathcal{X},\rho)}$ for binary classification adapting the proof of Theorem 3.8.

Theorem 3.14. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space constructed from the metric ρ. For the binary classification setting, the learning rule 2C1NN is weakly universally consistent for all processes $\mathbb{X} \in WSMV_{(\mathcal{X},\rho)}$.*

Proof Again, we follow a similar proof to that of Theorem 3.8. Let $\mathbb{X} \in \mathrm{WSMV}_{(\mathcal{X},\rho)}$ and consider the set $\mathcal{S}_{\mathbb{X}}$ of functions for which it is weakly consistent

$$\mathcal{S}_{\mathbb{X}} := \{A \in \mathcal{B}, \quad \mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, \mathbb{1}_A) \to 0\}.$$

By construction we have $\mathcal{S}_{\mathbb{X}} \subset \mathcal{B}$. The goal is to show that in fact $\mathcal{S}_{\mathbb{X}} = \mathcal{B}$. To do so, we will show that $\mathcal{S}$ satisfies the following properties

- $\emptyset \in \mathcal{S}_{\mathbb{X}}$ and $\mathcal{S}_{\mathbb{X}}$ contains all balls $B(x, r)$ with $x \in \mathcal{X}$ and $r \geq 0$,
- if $A \in \mathcal{S}_{\mathbb{X}}$ then $A^c \in \mathcal{S}_{\mathbb{X}}$ (stable to complementary),
- if $(A_i)_{i \geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$, then $\bigcup_{i \geq 1} A_i \in \mathcal{S}_{\mathbb{X}}$ (stable to $\sigma-$additivity for disjoint sets),
- if $A, B \in \mathcal{S}_{\mathbb{X}}$, then $A \cup B \in \mathcal{S}_{\mathbb{X}}$ (stable to union).

Together, these properties show that $\mathcal{S}_{\mathbb{X}}$ is a $\sigma-$algebra that contains all open intervals of $\mathcal{X}$. The invariance to complementary is again due to the fact that 2C1NN is invariant to relabeling. Further, we clearly have $\emptyset \in \mathcal{S}_{\mathbb{X}}$. Now let $x \in \mathcal{X}$ and $r \geq 0$, Proposition 3.3 shows that $B(x, r) \in \mathcal{S}_{\mathbb{X}}$.

We now turn to the $\sigma-$additivity for disjoint sets. Let $(A_i)_{i \geq 1}$ is a sequence of disjoint sets of $\mathcal{S}_{\mathbb{X}}$. We denote $A := \bigcup_{i \geq 1} A_i$. We consider the target function $f^* = \mathbb{1}_A$. We write the average loss in the following way,

$$\frac{1}{T} \sum_{t=1}^{T} \ell_{01}(2C1NN(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_t), f^*(X_t)) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{X_t \in A} \mathbb{1}_{X_{\phi(t)} \notin A} + \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A}.$$

We suppose by contradiction that 2C1NN is not weakly consistent on f^*. Then there exists $\epsilon > 0$ and an increasing sequence of times $(T_l)_{l \geq 1}$ such that $\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T_l) \geq \epsilon T_l$. We

first analyze the errors induced by one set A_i only. Simililarly to the proof of Theorem 3.8 we have

$$\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})\leq\frac{1}{T}\sum_{t=1}^{T}\ell_{01}(2C1NN(\mathbb{X}_{<t},\mathbb{1}_{\mathbb{X}_{<t}\in A_i},X_t),\mathbb{1}_{X_t\in A_i}).$$

Then, because 2C1NN is consistent for $\mathbb{1}_{.\in A_i}$, we get

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})\right]\to 0.$$

We take $\epsilon_i=\frac{\epsilon}{4\cdot 2^i}$ and T^i such that

$$\forall T\geq T^i,\quad \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})\right]<\frac{\epsilon_i^2}{2}.$$

We now consider the scale of the process $\mathbb{X}_{\leq T^i}$ when falling in A_i, by introducing $\eta_i>0$ such that

$$\mathbb{P}\left[\min_{\substack{t_1,t_2\leq T^i;\ X_{t_1},X_{t_2}\in A_i;\\ X_{t_1}\neq X_{t_2}}}\rho(X_{t_1},X_{t_2})>\eta_i\right]\geq 1-\frac{\epsilon_i}{2}.$$

We denote by $\mathcal{F}_i$ this event. Thus, $\mathbb{P}[\mathcal{F}_i^c]\leq\frac{\epsilon_i}{2}$. We now construct a partition $\mathcal{P}$ obtained by subdividing each set A_i according to scale η_i. Because $\mathcal{X}$ is separable, there exists a sequence of points $(x^j)_{j\geq 1}$ in $\mathcal{X}$ such that $\forall x\in\mathcal{X},\inf_{j\geq 1}\rho(x,x^j)=0$. We construct the following partition of $\mathcal{X}$ given by

$$\mathcal{P}\ :\quad A^c\cup\bigcup_{i\geq 1}\bigcup_{j\geq 1}\left\{\left(B\left(x^j,\frac{\eta_i}{2}\right)\cap A_i\right)\setminus\bigcup_{k<j}B\left(x^k,\frac{\eta_i}{2}\right)\right\}.$$

We now fix $l\geq 1$ and consider the event $\mathcal{A}:=\{\mathcal{L}_{\mathbb{X}}(2C1NN,f^*;T_l)\geq\frac{\epsilon}{2}\}$. Note that

$$\epsilon T_l\leq\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN,f^*;T_l)\leq\frac{\epsilon}{2}T_l+\mathbb{P}[\mathcal{A}]T_l,$$

which gives $\mathbb{P}[\mathcal{A}]\geq\frac{\epsilon}{2}$. We also define the following event

$$\mathcal{E}_i=\left\{\frac{1}{T_l}\sum_{t=1}^{T_l}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})<\epsilon_i\right\},$$

for any $i\in I:=\{i\geq 1,\ T_l\geq T^i\}$. Then, we have

$$\frac{\epsilon_i^2}{2}\geq\mathbb{E}\left[\frac{1}{T_l}\sum_{t=1}^{T_l}(\mathbb{1}_{X_t\in A_i}\mathbb{1}_{X_{\phi(t)}\notin A}+\mathbb{1}_{X_t\notin A}\mathbb{1}_{X_{\phi(t)}\in A_i})\right]\geq\epsilon_i\mathbb{P}[\mathcal{E}_i^c],$$

which yields $\mathbb{P}[\mathcal{E}_i^c]\leq\frac{\epsilon_i}{2}$. We will now focus on the event $\mathcal{A}\cap\bigcap_{i\in I}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i$, which has probability $\mathbb{P}[\mathcal{A}\cap\bigcap_{i\in I}\mathcal{E}_i\cap\bigcap_{i\geq 1}\mathcal{F}_i]\geq\mathbb{P}(\mathcal{A})-\sum_{i\in I}\mathbb{P}[\mathcal{E}_i^c]-\sum_{i\geq 1}\mathbb{P}[\mathcal{F}_i^c]\geq\frac{\epsilon}{2}-\frac{\epsilon}{4}=\frac{\epsilon}{4}$. Let us

now consider a realization of $\boldsymbol{x}$ of $\mathbb{X}$ in the event $\mathcal{A} \cap \bigcap_{i\in I} \mathcal{E}_i \cap \bigcap_{i\geq 1} \mathcal{F}_i$. The sequence $\boldsymbol{x}$ is now not random anymore. We will show that $\boldsymbol{x}$ does visits a linear number of sets in the partition $\mathcal{P}$.

Because the event $\mathcal{A}$ is met, we have

$$\sum_{i\geq 1} \frac{1}{T_l} \sum_{t=1}^{T_l} (\mathbb{1}_{x_t\in A_i} \mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A} \mathbb{1}_{x_{\phi(t)}\in A_i}) \geq \frac{\epsilon}{2}.$$

Also, because the events $\mathcal{E}_i$ are met, we have

$$\sum_{i\in I} \frac{1}{T_l} \sum_{t=1}^{T_l} (\mathbb{1}_{x_t\in A_i} \mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A} \mathbb{1}_{x_{\phi(t)}\in A_i}) \leq \sum_{i\in I} \epsilon_i \leq \frac{\epsilon}{4}.$$

Combining the two above equations gives

$$\frac{1}{T_l} \sum_{t=1}^{T_l} \sum_{i\notin I} (\mathbb{1}_{x_t\in A_i} \mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A} \mathbb{1}_{x_{\phi(t)}\in A_i}) > \frac{\epsilon}{4}. \tag{3.3}$$

We now consider the set of times such that an input point fell into the set A_i with $i \notin I$, either creating a mistake in the prediction of 4C1NN or inducing a later mistake within time horizon T_l: $\mathcal{T} := \bigcup_{i\notin I} \mathcal{T}_i$ where

$$\mathcal{T}_i := \left\{ t \leq T_l,\ x_t \in A_i,\ \left(x_{\phi(t)} \notin A \text{ or } \exists t < u \leq T_l \text{ s.t. } \phi(u) = t,\ x_u \notin A \right) \right\}.$$

Because the events $\mathcal{F}_i$ are met, the same arguments as in the proof of Theorem 3.8 show that all points x_t for $t \in \mathcal{T}$ fall in distinct sets of the partition $\mathcal{P}$, i.e. $|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}|$. We also obtain with the same arguments

$$\sum_{t=1}^{t_k} \sum_{i\notin I} (\mathbb{1}_{x_t\in A_i} \mathbb{1}_{x_{\phi(t)}\notin A} + \mathbb{1}_{x_t\notin A} \mathbb{1}_{x_{\phi(t)}\in A_i}) \leq 3|\mathcal{T}|.$$

We now use Eq (3.3) to obtain $|\{P \in \mathcal{P}, P \cap \boldsymbol{x}_{\leq t_k} \neq \emptyset\}| \geq |\mathcal{T}| \geq \frac{\epsilon}{12} T_l$. Therefore, because this holds for any realization in $\mathcal{A} \cap \bigcap_{i\in I} \mathcal{E}_i \cap \bigcap_{i\geq 1} \mathcal{F}_i$ we obtain

$$\mathbb{E}[|\{P \in \mathcal{P},\ P \cap \mathbb{X}_{\leq T_l} \neq \emptyset\}|] \geq \mathbb{P}\left[\mathcal{A} \cap \bigcap_{i\in I} \mathcal{E}_i \cap \bigcap_{i\geq 1} \mathcal{F}_i \right] \frac{\epsilon}{12} T_l \geq \frac{\epsilon^2}{48} T_l.$$

This holds for any $l \geq 1$. Therefore, because $(T_l)_{l\geq 1}$ is an increasing sequence, this shows that $\mathbb{X} \notin \mathrm{WSMV}_{(\mathcal{X},\rho)}$ which contradicts the hypothesis. This concludes the proof that $A \in \mathcal{S}_{\mathbb{X}}$ and hence, $\mathcal{S}_{\mathbb{X}}$ satisfies the disjoint $\sigma-$additivity property.

We now show that $\mathcal{S}_{\mathbb{X}}$ is invariant to finite unions. Let $A_1, A_2 \in \mathcal{S}_{\mathbb{X}}$. We consider $A = A_1 \cup A_2$ and $f^*(\cdot) = \mathbb{1}_{\cdot\in A}$. Using the same arguments as above, we still have for $T \geq 1$,

$$\mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} (\mathbb{1}_{X_t\in A_i} \mathbb{1}_{X_{\phi(t)}\notin A} + \mathbb{1}_{X_t\notin A} \mathbb{1}_{X_{\phi(t)}\in A_i}) \right] \to 0.$$

for $i \in \{1, 2\}$. But note that for any $T \geq 1$,

$$\begin{aligned}
\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) &= \sum_{t=1}^{T} \mathbb{1}_{X_t \in A} \mathbb{1}_{X_{\phi(t)} \notin A} + \sum_{t=1}^{T} \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A} \\
&\leq \sum_{t=1}^{T} (\mathbb{1}_{X_t \in A_1} + \mathbb{1}_{X_t \in A_2}) \mathbb{1}_{X_{\phi(t)} \notin A} + \sum_{t=1}^{T} \mathbb{1}_{X_t \notin A} (\mathbb{1}_{X_{\phi(t)} \in A_1} + \mathbb{1}_{X_{\phi(t)} \in A_2}) \\
&= \sum_{i=1}^{2} \sum_{t=1}^{T} (\mathbb{1}_{X_t \in A_i} \mathbb{1}_{X_{\phi(t)} \notin A} + \mathbb{1}_{X_t \notin A} \mathbb{1}_{X_{\phi(t)} \in A_i}).
\end{aligned}$$

Therefore we obtain directly $\mathbb{E}\left[\frac{1}{T}\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T)\right] \to 0$. This shows that $A_1 \cup A_2 \in \mathcal{S}_{\mathbb{X}}$ and ends the proof of the theorem. ■

Finally, we turn to the case of a bounded separable output setting $(\mathcal{Y}, \ell)$ and show that 2C1NN is weakly optimistically universal. In the case of weak learning, the reduction from any separable bounded output setting does not require a sophisticated argument as in the proof of Theorem 3.12, and can be made using the dominated convergence theorem.

Theorem 3.15. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space constructed from the metric ρ. The learning rule 2C1NN is weakly universally consistent for all processes $\mathbb{X} \in WSMV_{(\mathcal{X},\rho)}$ and any separable bounded output setting $(\mathcal{Y}, \ell)$.*

Proof We fix an output setting $(\mathcal{Y}, \ell)$ and let $\mathbb{X} \in \mathrm{WSMV}_{(\mathcal{X},\rho)}$. We will show that 2C1NN is weakly universally consistent on $\mathbb{X}$ for $(\mathcal{Y}, \ell)$.

We first start by showing that it is weakly universally consistent for classification with countable number of classes $(\mathbb{N}, \ell_{01})$. We fix a target function $f^* : \mathcal{X} \to \mathbb{N}$. For any $i \in \mathbb{N}$ we define the binary function $f_i^* := \mathbb{1}(f^*(\cdot) = i)$. We define

$$\mathcal{L}_i(T) := \sum_{t=1}^{T} \mathbb{1}_{f^*(x_t)=i} \ell_{01}(f^*(x_{\phi(t)}), f^*(x_t))$$

for all $i \geq 0$. Then,

$$\mathcal{L}_i(T) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{f^*(x_t)=i} \ell_{01}(f_i^*(x_{\phi(t)}), f_i^*(x_t)) \leq \mathcal{L}_{\mathbb{X}}(2C1NN, f_i^*; T)$$

Therefore, because 2C1NN is weakly universally consistent for binary classification from Theorem 3.14, we have $\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, f_i^*; T) \to 0$, hence $\mathbb{E}\mathcal{L}_i(T) \to 0$ for all $i \geq 0$. Since $\mathcal{L}_i(T) \geq 0$ and $\sum_{i \geq 0} \mathcal{L}_i(T) = \mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) \leq 1$, we can apply the dominated convergence theorem and obtain

$$\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) \to 0,$$

which proves that 2C1NN is weakly universally consistent for classification with countable number of classes.

We now turn to the general setting $(\mathcal{Y}, \ell)$. Let $(y^i)_{i\geq 1}$ be a a dense sequence on $\mathcal{Y}$ with respect to ℓ, let $\epsilon > 0$ and consider the function $h(y) := \inf\{i \geq 1 : \ \ell(y^i, y) < \epsilon\}$. Then, we have

$$\begin{aligned}\ell(y_{\phi(t)}, y_t) &\leq \bar{\ell} \cdot \mathbb{1}_{h(y_{\phi(t)}\neq h(y_t)} + \ell(y_{\phi(t),y_t})\mathbb{1}_{h(y_{\phi(t)}=h(y_t)} \\ &\leq \bar{\ell} \cdot \ell_{01}\mathbb{1}_{h\circ f^*(x_{\phi(t)})\neq h\circ f^*(x_t)} + c_\ell(\ell(y_{\phi(t)}, y^{h(y_{\phi(t)})}) + \ell(y^{h(y_t)}, y_t)) \\ &\leq \bar{\ell} \cdot \ell_{01}\mathbb{1}_{h\circ f^*(x_{\phi(t)})\neq h\circ f^*(x_t)} + 2c_\ell\epsilon.\end{aligned}$$

This yields $\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) \leq \bar{\ell}\mathcal{L}_{\mathbb{X}}(2C1NN, h \circ f^*; T) + 2c_\ell\epsilon$. Because 2C1NN is weakly universally consistent for countably-many classification, we have $\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, h \circ f^*; T) \to 0$. Therefore, we obtain

$$\limsup_T \mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) \leq 2c_\ell\epsilon.$$

This holds for any $\epsilon > 0$ therefore, $\mathbb{E}\mathcal{L}_{\mathbb{X}}(2C1NN, f^*; T) \to 0$, which ends the proof that 2C1NN is weakly universally consistent on $\mathbb{X}$ for the setting $(\mathcal{Y}, \ell)$. ■

As an immediate consequence, we have $\text{WSMV}_{(\mathcal{X},\rho)} \subset \text{WOUL}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)}$. Together with Proposition 3.1 we obtain a complete characterization for weak learnable processes and show that 2C1NN is weakly optimistically universal for general output value spaces.

Corollary 3.3. *For any separable Borel space $(\mathcal{X}, \mathcal{B})$, and every separable near metric space $(\mathcal{Y}, \ell)$ with $0 < \bar{\ell} < \infty$ we have $\text{WOUL}_{(\mathcal{X},\rho),(\mathcal{Y},\ell)} = \text{WSMV}_{(\mathcal{X},\rho)}$. In particular, SOUL is invariant from the output setup. Further, 2C1NN is weakly optimistically universal.*

This completely closes the main questions on universal online learning with bounded losses from [Han21a; Han21b] as we have now proved Theorem 3.2 and Theorem 3.3 (Corollaries 3.2 and 3.3).

3.9 Universal Learning with Unbounded Losses

3.9.1 Prior works on universal learning with bounded losses

In the case of *unbounded* losses the two main questions of (1) characterizing the family SOUL in terms of properties of the stochastic process $\mathbb{X}$, and (2), identifying particular learning rules that are optimistically universal, were already settled. Specifically, [Han21a] shows that, for any unbounded loss, there exists optimistically universal online learning rules. Moreover, [Han21a] also expresses a condition which characterizes the family SOUL. The condition requires that, for every countable measurable partition of $\mathcal{X}$, the process $\mathbb{X}$ visits a finite number of cells almost surely. This will be referred to as the "finite measurable visits" (FMV) condition:

Condition FMV. *For every disjoint sequence $\{A_k\}_{k=1}^\infty$ in $\mathcal{B}$ with $\cup_{k=1}^\infty A_k = \mathcal{X}$ (i.e., every countable measurable partition),*

$$\#\{k \in \mathbb{N} : A_k \cap \mathbb{X} \neq \emptyset\} < \infty \quad (a.s).$$

By abuse of notation, let FMV denote the set of all processes $\mathbb{X}$ satisfying this condition.

It is worth noting, however, that the specification and analysis of the optimistically universal learning rule in [Han21a], and the proof that Condition FMV indeed characterizes SOUL, are all quite complicated. For instance, the algorithm and its analysis directly rely on a construction of a countable dense subset of the set of measurable functions, under a metric appropriate to learning with unbounded losses. The learning algorithm then solves a sequence of constraint satisfaction problems specified in terms of this countable dense set of functions, to select one such function as its predictor. It is therefore desirable to *simplify* the theory, not only for practical reasons, but also to help us to intuitively understand the varieties of processes that admit universally consistent learners, and to clarify what kinds of learning rules can be optimistically universal. Toward this end, [Han21a] poses an important open question regarding a potential simplification: is SOUL characterized by Condition FS? The main contribution of the present section is showing that indeed this is *true* (Theorem 3.4). This fact allows us to dramatically simplify the entire theory of universally consistent online learning with unbounded losses. Several simplifications are immediate from this:

1. This provides a new, stronger characterization of SOUL.
2. The proof establishing FS = SOUL is significantly simpler than the original proof that FMV = SOUL.
3. The equivalence FS = SOUL immediately implies that the simple *memorization* rule is optimistically universal. This contrasts with the complicated construction used in the optimistically universal learner of [Han21a], which solves a sequence of constraint satisfaction problems in terms of a countable dense set of measurable functions.

At first glance, one might think that the class SOUL and its characterization should somehow depend on the specific setup given by $(\mathcal{X}, \rho)$, $\mathcal{Y}$ and ℓ. However, as [Han21a] showed (and as is implied by our Theorem 3.4), this dependency is very mild. In particular, the existence of an optimistically universal learning rule does not depend on the choice of $(\mathcal{Y}, \ell)$ as long as the loss is unbounded: $\sup_{y,y' \in \mathcal{Y}} \ell(y, y') = \infty$. Moreover, our results will hold for any separable metric space $(\mathcal{X}, \rho)$.

Outline of the proof. The essential strategy of the proof relies on the fact that FS $\subset$ SOUL $\subset$ FMV. The left inclusion is rather obvious. Indeed, if $\mathbb{X}$ contains a finite number of distinct values (a.s.), even the simple *memorization* learning rule is universally consistent. For the sake of thoroughness, we include a brief proof of this observation in Section 3.9.2. The second inclusion, that SOUL $\subset$ FMV, was shown by [Han21a] as part of the proof that SOUL = FMV. Given these inclusions FS $\subset$ SOUL $\subset$ FMV, what remains is establishing that FMV = FS. Establishing this equivalence is the main technical contribution of this section (Theorem 3.17). Its proof relies on constructing random measurable partitions of the space $\mathcal{X}$. We now turn to discussing the details of each of these components.

3.9.2 Sufficient and necessary condition for universally learnable processes

We begin with the easiest of the claimed inclusions: namely, FS $\subset$ SOUL. Recall that condition FS corresponds to having a finite number of values almost surely, i.e. $\#\mathbb{X} <$

∞ $(a.s.)$. While it may be rather obvious that all such processes admit a strong universal learning rule (i.e., they belong to SOUL), for the sake of thoroughness we present a simple proof of this fact.

Proposition 3.7. *FS $\subset$ SOUL. In particular, the memorization rule is universally consistent under every $\mathbb{X} \in$ FS.*

Proof We will show that the memorization learning rule is universally consistent under every $\mathbb{X}$ that takes a finite number of values almost surely. We can formally prove this result as follows. Let $\mathbb{X} \in$ FS be a given stochastic process and let $\{f_t\}_{t=1}^{\infty}$ be the memorization learning rule defined earlier. Observe that, for any measurable target function $f^* : \mathcal{X} \to \mathcal{Y}$, the (random) quantity $\mathrm{M} = \max_{t \in \mathbb{N}} \ell(y_0, f^*(X_t))$ is always finite (a.s.), as it is a maximum over a finite set: $\mathrm{M} < \infty$ $(a.s.)$. Now observe that the memorization rule makes at most $\#\mathbb{X}$ errors, each of value at most M. Therefore, $0 \leq \hat{\mathcal{L}}_{\mathbb{X}}(f_., f^*, T) \leq \frac{1}{T}\mathrm{M} \cdot \#\mathbb{X} \underset{T \to \infty}{\longrightarrow} 0, (a.s.)$. ■

An important remark is that this condition is not *testable*: there does not exists a consistent hypothesis test for FS. In other terms it is not possible to decide from the stream of input data $\mathbb{X}$ whether the process satisfies FS or not. Formally, a *hypothesis test* refers to a sequence of possibly random decision functions $\hat{t}_n : \mathcal{X} \to \{0, 1\}$. We then say that a test is *consistent* for a class of processes $\mathcal{C}$ if for any process $\mathbb{X}$, $\hat{t}_n(\mathbb{X}_{\leq n}) \to \mathbf{1}_{\mathbb{X} \in \mathcal{C}}$ in probability.

Proposition 3.8. *If $\mathcal{X}$ is infinite, there is no consistent hypothesis test for condition FS.*

Proof This is a consequence from Theorem 3.4 and the fact that there is no consistent hypothesis test for SOUL when $\mathcal{X}$ is infinite, shown in [Han21a, Theorem 59]. For completeness, we provide a direct and simplified proof below.

Since $\mathcal{X}$ infinite, let $\{x_i\}_{i \geq 0}$ a sequence of distinct points of $\mathcal{X}$. Let $\{\hat{t}_n\}_{n \geq 0}$ be a hypothesis test for condition FS. We suppose by contradiction that $\{\hat{t}_n\}$ is consistent and aim to construct a sequence $\mathbb{X}$ on which it fails. Following a proof construction introduced in [Han21a, Theorem 47], we construct a (deterministic) process which fools the test by alternatively switching between two modes: constant $X_t = x_0$ or visiting points of the distinct sequence $X_t = x_t$. Let $n_0 = 0$ and $X_0 = x_0$. We construct the sequences $\mathbb{X}$ and $(n_i)_{i \geq 0}$ by induction. Suppose we have constructed n_t for $0 \leq t \leq k-1$ and X_t for $0 \leq t \leq n_{k-1}$.

- If k is even, consider the deterministic process $\mathbb{Y}$ such that $Y_t = X_t$ for $t \leq n_{k-1}$ and $Y_t = x_0$ for $t > n_{k-1}$. Because $\{\hat{t}_n\}$ is consistent, we can define an index $n_k > n_{k-1}$ such that $\mathbb{P}(\hat{t}_{n_k}(\mathbb{Y}_{\leq n_k}) = 1) > \frac{3}{4}$.
- If k odd, consider the deterministic process $\mathbb{Y}$ such that $Y_t = X_t$ for $t \leq n_{k-1}$ and $Y_t = x_t$ for $t > n_{k-1}$. Similarly, let $n_k > n_{k-1}$ such that $\mathbb{P}(\hat{t}_{n_k}(\mathbb{Y}_{\leq n_k}) = 0) > \frac{3}{4}$.

We then set $X_t = Y_t$ for $n_{k-1} < n_k$. Note that for all $k \geq 0$, $\mathbb{P}(\hat{t}_{n_{2k}}(\mathbb{X}_{\leq n_{2k}}) = 1) > \frac{3}{4}$ and $\mathbb{P}(\hat{t}_{n_{2k+1}}(\mathbb{X}_{\leq n_{2k+1}}) = 1) < \frac{1}{4}$. Then, $\hat{t}_n(\mathbb{X}_{\leq n})$ does not converge in probability and the hypothesis test $\{\hat{t}_n\}$ is not consistent. This ends the proof of the proposition. ■

This justifies the terminology "optimistic" in *optimistically universal learning rule* in the sense that belonging to the set of sequences for which universal learning is achievable,

which we will prove is equal to FS, is a non-testable assumption. We now recall a necessary condition for a stochastic process $\mathbb{X}$ to admit strong universal online learning. It was shown that condition FMV, which requires that $\mathbb{X}$ will only visit a finite number of zones for all given countable measurable partitions of $\mathcal{X}$, is necessary for universal learning; we also include the proof for the sake of completeness.

Theorem 3.16 ([Han21a]). *SOUL $\subset$ FMV.*

Proof Let $\mathbb{X}$ be a stochastic process that does not satisfy FMV and f_n be a learning rule, we aim to show that this learning rule cannot be universally consistent. By hypothesis, there exists a finite measurable partition $\{A_k\}_{k=1}^{\infty}$ in $\mathcal{B}$ such that $\mathbb{X}$ visits an infinite number of the A_k with probability $p > 0$. We denote by $\mathcal{A}$ this event. We call $\mathcal{F}$ the class of measurable functions $f : \mathcal{X} \to \mathcal{Y}$ that takes constant values on each of the A_k. We will show that some objective function $f^* \in \mathcal{F}$ cannot be learnt by f_n.

First, let us define τ_k the first instant at which $\mathbb{X}$ attains A_k.

$$\tau_k = \begin{cases} \min\{t \in \mathbb{N} : X_t \in A_k\} & \text{if } A_k \cap \mathbb{X} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

We also define a deterministic quantity $T_k \in \mathbb{N}$ that upper bounds the τ_k with high probability, i.e.

$$\mathbb{P}(\tau_k \leq T_k) > 1 - 2^{-k}, \quad \forall k \geq 1.$$

By the Borel Cantelli lemma, since $\sum_k \mathbb{P}(\tau_k > T_k) < \infty$, almost surely there exists $\kappa \in \mathbb{N}$ such that $\tau_k \leq T_k$ for $k \geq \kappa$. We will denote $\mathcal{E}$ this event. We now sample f^* randomly from $\mathcal{F}$ as follows:

$$f^*(x \in A_k) = \begin{cases} y_{k,0} & \text{with proba } 1/2, \\ y_{k,1} & \text{with proba } 1/2, \end{cases}$$

where $y_{k,1}$ and $y_{k,0}$ are selected such that $\ell(y_{k,1}, y_{k,0}) \geq 2c_\ell T_k$ (recalling that c_ℓ denotes the constant from the relaxed triangle inequality satisfied by ℓ). Note that taking the expectation over the randomness in f^* allows to write:

$$\sup_{g \in \mathcal{F}} \mathbb{E}_{\mathbb{X}}(\mathcal{L}_{\mathbb{X}}(f_\cdot, g)) \geq \mathbb{E}_{f^*,\mathbb{X}}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*)). \tag{3.4}$$

We first prove a lower bound on the right term. Conditionally on $\mathcal{A} \cap \mathcal{E}$, observe that for any $k \geq \kappa$, $(\mathbb{X}_{<\tau_k}, \mathbb{Y}_{<\tau_k})$ provides no information on $f^*(X_{\tau_k})$. Then, the average of the corresponding prediction error satisfies $\mathbb{E}(\ell(f_{\tau_k}(\mathbb{X}_{<\tau_k}, \mathbb{Y}_{<\tau_k}, X_{\tau_k}), f^*(X_{\tau_k})) \geq T_k \geq \tau_k$, where we used the fact that ℓ satisfies the relaxed triangle inequality. Thus, in $\mathcal{A} \cap \mathcal{E}$, $\mathbb{E}_{f^*}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*, \tau_k)) \geq 1$ for any $k \geq \kappa$, hence by Fatou's lemma

$$\mathbb{E}_{f^*}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*)) \geq \limsup_{t \in \mathbb{N}} \mathbb{E}_{f^*}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*, \tau_k)) \geq 1.$$

Therefore, $\mathbb{P}_{\mathbb{X}}(\mathbb{E}_{f^*}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*)) \geq 1) \geq \mathbb{P}(\mathcal{A} \cap \mathcal{E}) = p$, which yields $\mathbb{E}_{f^*,\mathbb{X}}(\mathcal{L}_{\mathbb{X}}(f_\cdot, f^*)) \geq p$. Eq (3.4) then shows that there exists $g \in \mathcal{F}$ such that $\mathbb{E}_{\mathbb{X}}(\mathcal{L}_{\mathbb{X}}(f_\cdot, g)) > 0$, hence $\mathcal{L}_{\mathbb{X}}(f_\cdot, g) > 0$ with nonzero probability. This ends the proof of the result. ■

In summary, we have the inclusions FS $\subset$ SOUL $\subset$ FMV. This allows for a concise expression of the conjecture formulated by [Han21a] (which, in light of the result of [Han21a] that SOUL = FMV, is equivalent to the formulation of the problem stated earlier).

Question 3.5 ([Han21a]). *Is it true that FMV = FS?*

We will prove that this equality holds for all separable metric spaces $\mathcal{X}$. This in turn implies that SOUL = FS and therefore ensures that "memorization", which we already saw is universally consistent for all processes in FS (Proposition 3.7), is an optimistically universal learning rule. The solution to the question will be detailed in Section 3.9.3 and generalised to all separable metric spaces in Section 3.9.4. We conclude this section by giving some additional inspiration for the proofs that will follow.

Remarks on Question 3.5. In words, the question asked is whether the set of countable measurable partitions is sufficiently large to separate all stochastic processes that take an infinite number of values.

It was already observed by [Han21a] that when $\mathcal{X}$ is countable or when $\mathbb{X}$ is deterministic, FS and FMV are equal. However, both these setups come with a natural partition: if $\mathcal{X}$ is countable, $\{\{x\} : x \in \mathcal{X}\}$ becomes a countable measurable partition of $\mathcal{X}$, and when $\mathbb{X}$ is deterministic $\{\{x\} : \{x\} \cap \mathbb{X} \neq \emptyset\} \cup \{\mathcal{X} \setminus \mathbb{X}\}$ will also isolate all the different values taken by $\mathbb{X}$.

In the uncountable case, for instance when $\mathcal{X} = \mathbb{R}$, we aim to define a partition $\{A_k\}_{k=1}^{\infty}$ that scatters the space. We want to minimize the chance that two values taken by the process, say $X_t \neq X_{t'}$, fall in the same A_k. A classical and tempting way to build such a partition would be using the axiom of choice [Vit05; Ste85]. Define the equivalence relation $x \sim_{\mathbb{Q}} y \iff x - y \in \mathbb{Q}$, where $\mathbb{Q}$ can be enumerated $\mathbb{Q} = \{q_1, q_2, ...\}$. Now for each of the equivalence classes of the form $\{x\} + \mathbb{Q}$, *choose* one representer. Denote by A the set of all representers and observe that $\{A + \{q\}\}_{q\in\mathbb{Q}}$ makes a countable partition of $\mathbb{R}$. Note that two different values of $\mathbb{X}$, say $X_t \neq X_{t'}$, fall in the same equivalence class only if $X_t - X_{t'}$ was chosen as a representer. This event could be made very rare if we were to shift all representers by a uniform random variable, or to choose the representer at random within their class of equivalence. The reason why this does not prove the result is that the corresponding partition $\{A_k\}_{k=1}^{\infty}$ is not measurable.

Another idea to create such a random partition $\{A_k\}_{k=1}^{\infty}$ would be to assign each $x \in \mathbb{R}$ to a set $A_{k(x)}$ where the index $k(x) \in \mathbb{N}$ is chosen independently at random following an exponential law $\mathcal{E}(\frac{1}{2})$: $\mathbb{P}(k(x) = k) = \frac{1}{2^k}$. The indices $\{k(X_1), k(X_2), ...\}$ to which the elements of the sequence $\mathbb{X} = \{X_1, X_2, ...\}$ are assigned, are almost surely unbounded when $\#\mathbb{X} = +\infty$, disproving condition FMV. We will refer to this construction as the partition $\mathcal{P}$ as it will later be a useful inspiration. Unfortunately, as such, $\mathcal{P}$ not define a proper partition because the sets A_k are not measurable in general. To solve this issue, instead of defining point-wise random sets, we will use countable union of small intervals. Depending on the scale of the process $\mathbb{X}$, these sets will give same behaviour as the parts of $\mathcal{P}$. We will make this idea more precise in the following paragraph.

We first recall a construction of dense open sets of $\mathbb{R}$ with measure at most $\epsilon > 0$. Following a classical argument, one can consider the union of open intervals $\cup_{i\geq 1} \left(q_i - \frac{\epsilon}{2^i}, q_i + \frac{\epsilon}{2^i}\right)$,

where $\{q_1, q_2, ...\}$ are i.i.d. sampled from some probability density of full support. If we denote the remainders $R_k = \cup_{i \geq k} \left(q_i - \frac{\epsilon}{2^i}, q_i + \frac{\epsilon}{2^i}\right)$ and consider the partition $\{A_k\}_{k=0}^{\infty}$ defined by $A_k = R_k \setminus R_{k+1}$ where $R_0 = \mathbb{R}$, one could hope that any sequence $\mathbb{X}$ taking infinite values will visit an infinite number of the A_k. In fact, this is true if the convergence rate of $\mathbb{X}$ is not too fast but not in the general case. We will therefore use a decay rate adapted to the process $\mathbb{X}$ through a parameter δ_k defined as follows,

$$\delta_k = \min\left\{ |x - y| \middle| x, y \in \mathbb{X}_{\leq N}, \#\mathbb{X}_{\leq N} \geq 2^{2k+2} \right\}.$$

A key intuition is that the first k distinct points visited by $\mathbb{X}$ have scale δ_k. Thus, a remainder R_i of smaller scale – such that the length of the intervals defining R_i is $\ll \delta_k$ – will appear uniformly random to the first k distinct inputs, similarly to the point-wise random sets from the partition $\mathcal{P}$ introduced above.

Technical result. We prove the equivalence of FS and FMV therefore guaranteeing that memorization is an optimistically universal learning rule in the unbounded setup. The following result represents the technical contribution of this section.

Theorem 3.17. *For any separable metric space $(\mathcal{X}, \rho)$, FMV = FS.*

Together with Proposition 3.7 and Theorem 3.16, this implies the main Theorem 3.4.

3.9.3 Proof of the characterization for $\mathcal{X} = [0, 1]$

In this section, we prove the result for $\mathcal{X} = [0, 1]$, as it provides a direct and simple construction. The proof will then be generalised to all separable metric spaces in Section 3.9.4.

Proposition 3.9. *If $\mathcal{X} = [0, 1]$ with its usual topology, FMV = FS.*

Proof The inclusion FS $\subset$ FMV is a direct observation, therefore we focus on proving that FMV $\subset$ FS. Let $\mathbb{X}$ be a stochastic process which does not satisfy FS. The goal is to construct a countable measurable partition $\{A_k\}_{k=1}^{\infty}$ of $\mathcal{X}$ which disproves condition FMV, i.e. such that $\{k \in \mathbb{N} : A_k \cap \mathbb{X} \neq \emptyset\}$ is infinite with nonzero probability. Denote by $\mathcal{A}$ the event that $\mathbb{X}$ takes an infinite number of values, i.e. $\mathcal{A} = \{\#\mathbb{X} = +\infty\}$. We have assumed that $\mathbb{P}(\mathcal{A}) > 0$ and will condition on $\mathcal{A}$ for the rest of the proof. For $k \in \mathbb{N}$, define $N_k \in \mathbb{N}$ such that,

$$\mathbb{P}\left(\#\mathbb{X}_{\leq N_k} \geq \frac{1}{\mu_k^2} \middle| \mathcal{A} \right) \geq 1 - \frac{1}{2^{k+1}}, \quad \text{where} \quad \mu_k := \frac{1}{2^{k+1}}.$$

Note that N_k is a deterministic quantity that only depends on the process $\mathbb{X}$. It is well defined because $\mathbb{P}\left(\#\mathbb{X}_{\leq N} \geq \frac{1}{\mu_k^2} \middle| \mathcal{A} \right) \to_{N \to \infty} 1$ since $\mathbb{X}$ takes an infinite number of values in $\mathcal{A}$. Now also define $0 < \delta_k < \frac{1}{2^{k+1}}$ satisfying:

$$\mathbb{P}\left(\min_{1 \leq i,j \leq N_k, X_i \neq X_j} |X_i - X_j| > \delta_k \middle| \mathcal{A} \right) \geq 1 - \frac{1}{2^{k+1}}.$$

Let $\mathcal{E}_k$ be the intersection of the two events above, we have by union bound: $\mathbb{P}(\mathcal{E}_k \mid \mathcal{A}) \geq 1 - \frac{1}{2^k}$, where $\mathcal{E}_k$ can be written as

$$\mathcal{E}_k : \quad \#\mathbb{X}_{\leq N_k} > \frac{1}{\mu_k^2} \quad \text{and} \quad \forall x \neq y \in \mathbb{X}_{\leq N_k},\ |x - y| > \delta_k.$$

We are now ready to construct the partition. Let $\boldsymbol{q} = (q_i)_{i \geq 1}$ be an i.i.d. sequence of independent uniforms sampled from $\mathcal{U}([0,1])$. Define $B_0 = [0,1]$ and for $k \geq 1$,

$$B_k = \bigcup_{i_{k-1} < i \leq i_k} \left[q_i - \frac{\delta_k}{2}, q_i + \frac{\delta_k}{2} \right],$$

where $(i_k)_{k \geq 0}$ satisfies $i_0 = 0$ and $i_k = i_{k-1} + \lceil \frac{\mu_k}{\delta_k} \rceil$. Note that the Borel measure of B_k is roughly μ_k: $\mu(B_k) \leq \mu_k + \delta_k$. We use the remainders $R_k = [0,1] \cap \bigcup_{l \geq k} B_l$ to define the partition $\{A_k\}_{k=-1}^{\infty}$ as follows:

$$A_k = R_k \setminus R_{k+1}, \quad k \geq 0$$

and $A_{-1} = \bigcap_{k \geq 0} R_k$. The sets $\{A_k\}_{k=-1}^{\infty}$ define a proper partition of $\mathcal{X}$ since A_{-1} contains elements that appear infinitely often in $\{B_k\}_{k \geq 0}$ while for any $k \geq 0$, A_k contains elements that appear for the last time in the sequence $\{B_l\}_{k \geq 0}$ in B_k. This covers the whole space $\mathcal{X}$ because by construction $B_0 = \mathcal{X}$. The interest of this (random) construction lies in the two following lemmas.

Lemma 3.5. *For any finite (deterministic) $S \subset [0,1]$ with $\#S > \frac{1}{\mu_k^2}$ and $\forall x \neq y \in S, |x - y| > \delta_k$,*

$$\mathbb{P}(B_k \cap S = \emptyset) \leq e^{-2^{k+1}}.$$

Lemma 3.6. *For any countable (deterministic) $S \subset [0,1]$, $A_{-1} \cap S = \emptyset$, (a.s.)*

We will now show that with probability $\mathbb{P}(\mathcal{A})$, the partition $\{A_k\}_{k=-1}^{\infty}$ disproves the condition FMV, in other terms that $\mathbb{X}$ visits an infinite number of sets of the partition. Recall that the randomness is now both in terms of the stochastic process $\mathbb{X}$ and the partition generated from $\boldsymbol{q}$. We have that,

$$\mathbb{P}(B_k \cap \mathbb{X} = \emptyset \mid \mathcal{A}) \leq (1 - \mathbb{P}(\mathcal{E}_k | \mathcal{A})) + \mathbb{P}(B_k \cap \mathbb{X}_{\leq N_k} = \emptyset \mid \mathcal{E}_k, \mathcal{A}) \leq \frac{1}{2^k} + e^{-2^{k+1}},$$

where in the last inequality we applied Lemma 3.5 to the set $\mathbb{X}_{\leq N_k}$ which has cardinality at least $\frac{1}{\mu_k^2}$ in $\mathcal{E}_k$. We can now apply the first Borel-Cantelli lemma to the sequence of events $\{B_k \cap \mathbb{X} = \emptyset\}$ conditionally on $\mathcal{A}$, which shows that almost surely only a finite number of these events are satisfied. Hence, conditionally on $\mathcal{A}$, there exists almost surely $\kappa \in \mathbb{N}$ such that for every $k \geq \kappa$, the sequence $\mathbb{X}$ visits B_k. Further, by Lemma 3.6, with probability 1, $\mathbb{X}$ does not visit A_{-1}. Therefore, conditionnally on $\mathcal{A}$, the sequence almost surely visits an infinite number of sets of the partition $\{A_k\}_{k=-1}^{\infty}$. In summary,

$$\mathbb{P}_{\boldsymbol{q},\mathbb{X}}(\#\{k \in \mathbb{N} : A_k \cap \mathbb{X} \neq \emptyset\} = +\infty) \geq \mathbb{P}(\mathcal{A}).$$

Thus, there exists a *deterministic* choice of $\boldsymbol{q}$ yielding a partition $\{A_k\}_{k=-1}^{\infty}$ such that:

$$\mathbb{P}_{\mathbb{X}}(\#\{k \in \mathbb{N}, A_k \cap \mathbb{X} \neq \emptyset\} = +\infty) \geq \mathbb{P}(\mathcal{A}) > 0.$$

This shows the claim of the theorem. ■

It remains to give the missing proof of the two Lemmas 3.5 and 3.6.

Proof of Lemma 3.5 Note that the randomness of $\mathbb{X}$ does not intervene in this lemma. The probability law $\mathbb{P}$ only accounts for the randomness of the partition through the variables $\boldsymbol{q} = (q_i)_{i\geq 1}$. We enumerate $S = \{x_1, ..., x_T\}$ where $T = \#S \geq \frac{1}{\mu_k^2}$.

$$\mathbb{P}(B_k \cap S = \emptyset) = \mathbb{P}(x_1 \notin B_k) \prod_{t=2}^{T} \mathbb{P}(x_t \notin B_k | x_1, \ldots, x_{t-1} \notin B_k).$$

For the sake of simplicity we will use the notation $B(x, \delta) = [x - \delta, x + \delta]$. Note that events $\{x_i \notin B_k\}$ are negatively correlated. Indeed,

$$\begin{aligned}\mathbb{P}(x_t \notin B_k \mid x_1, \ldots, x_{t-1} \notin B_k) &= \prod_{i=i_{k-1}+1}^{i_k} \mathbb{P}\left[q_i \notin B\left(x_t, \frac{\delta_k}{2}\right) \middle| q_i \notin \bigcup_{1\leq l\leq t-1} B\left(x_l, \frac{\delta_k}{2}\right)\right] \\ &= \prod_{i=i_{k-1}+1}^{i_k} \mathbb{P}\left[\tilde{q}_i \notin B\left(x_t, \frac{\delta_k}{2}\right)\right]\end{aligned}$$

where $\tilde{q}_i \sim \mathcal{U}(J)$ with $J := \mathcal{X} \setminus \bigcup_{1\leq l\leq t-1} B(x_l, \frac{\delta_k}{2})$. Because $|x_t - x_l| > \delta_k$ for all $1 \leq l \leq t-1$, we have $B(x_t, \frac{\delta_k}{2}) \subset J$. Thus,

$$\mathbb{P}(x_t \notin B_k | x_1, \ldots, x_{t-1} \notin B_k) = \prod_{i=i_{k-1}+1}^{i_k} \left(1 - \frac{\delta_k}{\mu(J)}\right) \leq \prod_{i=i_{k-1}+1}^{i_k} (1 - \delta_k) = \mathbb{P}(x_t \notin B_k).$$

Using the negative correlation, we have that

$$\mathbb{P}(B_k \cap S = \emptyset) \leq \prod_{t=1}^{T} \mathbb{P}(x_1 \notin B_k) = (1 - \delta_k)^{T(i_k - i_{k-1})} \leq (1 - \delta_k)^{\frac{1}{\mu_k \delta_k}} \leq e^{-\frac{1}{\mu_k}} = e^{-2^{k+1}}.$$

This ends the proof of the lemma. ■

Proof of Lemma 3.6 We start by proving that for a given $x \in \mathbb{R}$, $x \notin A_{-1}$ a.s. For $k \geq 1$ we have,

$$\mathbb{P}(x \in B_k) \leq \left\lceil \frac{\mu_k}{\delta_k} \right\rceil \delta_k \leq \left(\frac{\mu_k}{\delta_k} + 1\right) \delta_k \leq \mu_k + \delta_k \leq \frac{1}{2^k}.$$

Therefore, $\mathbb{P}(x \in R_k) \leq \frac{1}{2^{k-1}}$. This shows that $\mathbb{P}(x \in A_{-1}) \leq \mathbb{P}(x \in \bigcap_k R_k) = 0$. Taking the union over all countable random variables in S, we have $\mathbb{P}(A_{-1} \cap S \neq \emptyset) = 0$. ■

Extension to all standard Borel spaces. Before we move on to proving the main theorem in the most general framework of separable metric spaces, observe that the proof for $\mathcal{X} = [0,1]$ easily extends to all standard Borel space by Kuratowski's theorem, in particular for instance to $\mathcal{X} = \mathbb{R}^d$. Kuratowski's theorem states that if $\mathcal{X}$ is an uncountable standard Borel space it is isomorphic to $[0,1]$ with the Euclidean distance, meaning that there exists a measurable bijection $f : \mathcal{X} \to [0,1]$. Let $\mathbb{X} = (X_i)_{i\geq 0}$ be a stochastic process on $\mathcal{X}$ satisfying FMV. Then, because f is measurable, $\tilde{\mathbb{X}} := (f(X_i))_{i\geq 0}$ is a stochastic process on $[0,1]$ which satisfies FMV. By Proposition 3.9, $\tilde{\mathbb{X}}$ satisfies FS. Thus, because f is bijective, $\mathbb{X}$ also satisfies FS.

3.9.4 Extension to General Separable Metric Spaces

The original proof of SOUL = FMV by [Han21a] holds for any separable metric space $(\mathcal{X}, \rho)$. In this section, we extend the proof above to hold in this more-general case as well, thus completely answering the question from Question 3.5 in full generality, and completing the proof of Theorem 3.4. For the remainder of this section, we let $(\mathcal{X}, \rho)$ denote a non-empty separable metric space, and we take as the set $\mathcal{B}$ of measurable subsets of $\mathcal{X}$ the Borel σ-algebra generated by the topology induced by ρ.

Theorem 3.17 (Restated) For any separable metric space $(\mathcal{X}, \rho)$, FMV = FS.

The main components of the proof are analogous to those for standard Borel spaces, with a few important changes: most importantly, the following lemma.

Lemma 3.7. *For any $\mathbb{X}$ satisfying condition FMV, for any $\delta, \epsilon > 0$ and $m_0 \in \mathbb{N}$, there exists $M_{\epsilon,\delta} \in \mathbb{N}$ with $M_{\epsilon,\delta} \geq m_0$, and a sequence $\mathcal{G}^{\epsilon,\delta} = \{G_1^{\epsilon,\delta}, \dots, G_{M_{\epsilon,\delta}}^{\epsilon,\delta}\}$ in $\mathcal{B}$ such that every distinct $i, j \in \{1, \dots, M_{\epsilon,\delta}\}$ satisfy $G_i^{\epsilon,\delta} \cap G_j^{\epsilon,\delta} = \emptyset$, and every $i \in \{1, \dots, M_{\epsilon,\delta}\}$ satisfies $\sup_{x,x' \in G_i^{\epsilon,\delta}} \rho(x, x') \leq \delta$, and such that*

$$\mathbb{P}\left(\mathbb{X} \cap \left(\mathcal{X} \setminus \bigcup_{i=1}^{M_{\epsilon,\delta}} G_i^{\epsilon,\delta}\right) \neq \emptyset\right) < \epsilon.$$

In other words, $\mathcal{G}^{\epsilon,\delta}$ is a sequence of disjoint measurable sets of diameter at most δ, which cover all of the points in $\mathbb{X}$ with probability $1 - \epsilon$.

Proof Let $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ be a countable dense subset: that is, $\sup_{x\in\mathcal{X}} \inf_{\tilde{x}\in\tilde{\mathcal{X}}} \rho(\tilde{x}, x) = 0$. Enumerate $\tilde{\mathcal{X}}$ as $\{\tilde{x}_1, \tilde{x}_2, \dots, \}$. Let $G_1^{\epsilon,\delta} = \{x : \rho(x, \tilde{x}_1) \leq \delta/2\}$, and for integers $k \geq 2$ inductively define $G_k^{\epsilon,\delta} = \{x : \rho(x, \tilde{x}_k) \leq \delta/2\} \setminus \bigcup_{k'=1}^{k-1} G_{k'}^{\epsilon,\delta}$. In particular, this collection $\{G_k^{\epsilon,\delta} : k \in \mathbb{N}\}$ forms a countable partition of $\mathcal{X}$ into measurable subsets of diameter at most δ (by the triangle inequality). Now let $\mathbb{X}$ be any process satisfying FMV. It remains only to show there exists a finite $M_{\epsilon,\delta} \in \mathbb{N}$ satisfying the claim. Let $\hat{M} = \max\{k : \mathbb{X} \cap G_k^{\epsilon,\delta} \neq \emptyset\}$, or $\hat{M} = \infty$ if there is no maximum. By hypothesis, $\mathbb{P}(\hat{M} < \infty) = 1$. Since the event $\{\hat{M} > M\}$ is non-increasing in M, $\lim_{M\to\infty} \mathbb{P}(\hat{M} > M) = \mathbb{P}(\hat{M} = \infty) = 0$. Thus, $\exists M_{\epsilon,\delta} \in \mathbb{N}$ with $M_{\epsilon,\delta} \geq m_0$ such that $\mathbb{P}(\hat{M} > M_{\epsilon,\delta}) < \epsilon$. In other words, $\mathbb{P}(\exists k > M_{\epsilon,\delta} : \mathbb{X} \cap G_k^{\epsilon,\delta} \neq \emptyset) < \epsilon$. Since $\{G_k^{\epsilon,\delta} : k \in \mathbb{N}\}$ is a partition of $\mathcal{X}$, this implies the claim in the lemma. ■

We are now ready for the main proof.

Proof of Theorem 3.17 Since condition FS clearly implies condition FMV, we focus on showing FMV $\subset$ FS. Let $\mathbb{X}$ be any process satisfying condition FMV, and for the sake of obtaining a contradiction, suppose that condition FS fails: that is, there is an event $\mathcal{A}$ with $\mathbb{P}(\mathcal{A}) > 0$, on which $\#\{x \in \mathcal{X} : \mathbb{X} \cap \{x\} \neq \emptyset\} = \infty$.

For each $k \in \mathbb{N}$, let $N_k \in \mathbb{N}$ be such that

$$\mathbb{P}\big(\#\mathbb{X}_{\leq N_k} \geq 2^{2k+2} \big| \mathcal{A}\big) \geq 1 - \frac{1}{2^{k+2}},$$

and let $\delta_k > 0$ be such that

$$\mathbb{P}\left(\min_{i,j \leq N_k : X_i \neq X_j} \rho(X_i, X_j) > \delta_k \middle| \mathcal{A}\right) \geq 1 - \frac{1}{2^{k+3}}.$$

Let $S_k = \{x \in \mathcal{X} : \mathbb{X}_{\leq N_k} \cap \{x\} \neq \emptyset\}$ and let $\epsilon_k = \frac{1}{2^{k+3}}$. Let $\mathcal{G}^{\epsilon_k,\delta_k}$ and M_{ϵ_k,δ_k} be as in Lemma 3.7, with $m_0 = 2^{k+2}$.

Let $\mathcal{E}_k$ denote the event that $\#\mathbb{X}_{\leq N_k} \geq 2^{2k+2}$, $\min_{i,j \leq N_k : X_i \neq X_j} \rho(X_i, X_j) > \delta_k$, and $\mathbb{X} \cap \big(\mathcal{X} \setminus \bigcup \mathcal{G}^{\epsilon_k,\delta_k}\big) = \emptyset$ all hold simultaneously. In particular, by the union bound, $\mathbb{P}(\mathcal{E}_k | \mathcal{A}) \geq 1 - 2^{-k-1}$.

For each $k \in \mathbb{N}$, let $b_k = \lceil 2^{-k-2} M_{\epsilon_k,\delta_k} \rceil$, and let $Q_1^k, \ldots, Q_{b_k}^k$ be independent uniform samples from $\mathcal{G}^{\epsilon_k,\delta_k}$ (also independent across k and independent from $\mathbb{X}$). Then let $B_k = \bigcup_{i=1}^{b_k} Q_i^k$. For each $k \in \mathbb{N}$, let $R_k = \bigcup_{\ell \geq k} B_\ell$. Also let $A_{-1} = \bigcap_{k \in \mathbb{N}} R_k$ and for each $k \in \mathbb{N}$, let $A_k = R_k \setminus R_{k+1}$, and $A_0 = \mathcal{X} \setminus R_1$. We will show that (with non-zero probability) the countable measurable partition $\{A_k : k \in \mathbb{N} \cup \{-1, 0\}\}$ violates the condition FMV, thus obtaining a contradiction.

Now note that, on the event $\mathcal{E}_k$, every $x \in S_k$ is in a distinct set $G_i^{\epsilon_k,\delta_k} \in \mathcal{G}^{\epsilon_k,\delta_k}$: that is, by definition of $\mathcal{E}_k$, every $x \in S_k$ is in some $G_i^{\epsilon_k,\delta_k} \in \mathcal{G}^{\epsilon_k,\delta_k}$, and since each $G_i^{\epsilon_k,\delta_k}$ has diameter at most δ_k, while every distinct $x, x' \in S_k$ are δ_k-separated (on event $\mathcal{E}_k$), no two elements of S_k can be in the same $G_i^{\epsilon_k,\delta_k}$. Therefore, on the event $\mathcal{E}_k$ we have that

$$\mathbb{P}(B_k \cap S_k = \emptyset | \mathbb{X}) = \mathbb{P}\big(Q_1^k \cap S_k = \emptyset \big| \mathbb{X}\big)^{b_k} = \left(1 - \frac{|S_k|}{M_{\epsilon_k,\delta_k}}\right)^{b_k} \leq e^{-|S_k| b_k / M_{\epsilon_k,\delta_k}} \leq e^{-2^k},$$

where the last inequality is based on the definition of b_k and the fact that $|S_k| \geq 2^{2k+2}$ on the event $\mathcal{E}_k$. Thus, on the event $\bigcap_{k \in \mathbb{N}} \mathcal{E}_k$,

$$\sum_{k=1}^{\infty} \mathbb{P}(B_k \cap S_k = \emptyset | \mathbb{X}) \leq \sum_{k=1}^{\infty} e^{-2^k} < \infty.$$

By the Borel-Cantelli lemma, this implies that there is an event $\mathcal{E}'$ of probability one, such that on $\mathcal{E}' \cap \bigcap_{k \in \mathbb{N}} \mathcal{E}_k$, there exists $\kappa \in \mathbb{N}$ such that every $k \geq \kappa$ satisfies $B_k \cap S_k \neq \emptyset$, and hence also $\mathbb{X} \cap R_k \neq \emptyset$. Now, if $\mathbb{X} \cap A_{-1} = \emptyset$, this would further imply that $|\{k \in \mathbb{N} : \mathbb{X} \cap A_k \neq \emptyset\}| = \infty$.

We next turn to showing that $\mathbb{X} \cap A_{-1} = \emptyset$ (a.s.). For any $t, k \in \mathbb{N}$, by the union bound, $\mathbb{P}(X_t \in B_k) \leq \frac{b_k}{M_{\epsilon_k,\delta_k}} \leq 2^{-k-1}$ (recalling that $M_{\epsilon_k,\delta_k} \geq 2^{k+2}$, so that $b_k \leq 2^{-k-1} M_{\epsilon_k,\delta_k}$). By

the union bound, this further implies any $t, k \in \mathbb{N}$ satisfy $\mathbb{P}(X_t \in R_k) \leq \sum_{\ell \geq k} \mathbb{P}(X_t \in B_\ell) \leq \sum_{\ell \geq k} 2^{-\ell-1} = 2^{-k}$. Thus, $\mathbb{P}(X_t \in A_{-1}) = \mathbb{P}\big(X_t \in \bigcap_{k\in\mathbb{N}} R_k\big) \leq \lim_{k\to\infty} \mathbb{P}(X_t \in R_k) = 0$. By the union bound, $\mathbb{P}(\mathbb{X} \cap A_{-1} \neq \emptyset) = 0$. Thus, there is an event $\mathcal{E}''$ of probability one, on which $\mathbb{X} \cap A_{-1} = \emptyset$.

Altogether, we have that on the event $\mathcal{E}' \cap \mathcal{E}'' \cap \bigcap_{k\in\mathbb{N}} \mathcal{E}_k$, $|\{k \in \mathbb{N} : \mathbb{X} \cap A_k \neq \emptyset\}| = \infty$. Since $\mathbb{P}(\mathcal{E}') = \mathbb{P}(\mathcal{E}'') = 1$, and

$$\begin{aligned}\mathbb{P}\left(\bigcap_{k\in\mathbb{N}} \mathcal{E}_k\right) &\geq \mathbb{P}\left(\mathcal{A} \cap \bigcap_{k\in\mathbb{N}} \mathcal{E}_k\right) \geq \mathbb{P}(\mathcal{A}) - \sum_{k\in\mathbb{N}} \mathbb{P}(\mathcal{A})\left(1 - \mathbb{P}(\mathcal{E}_k|\mathcal{A})\right) \\ &\geq \mathbb{P}(\mathcal{A}) - \sum_{k\in\mathbb{N}} \mathbb{P}(\mathcal{A}) 2^{-k-1} = \frac{1}{2}\mathbb{P}(\mathcal{A}),\end{aligned}$$

by the union bound we have $\mathbb{P}\big(\mathcal{E}' \cap \mathcal{E}'' \cap \bigcap_{k\in\mathbb{N}} \mathcal{E}_k\big) \geq \frac{1}{2}\mathbb{P}(\mathcal{A}) > 0$. In particular, this implies $\mathbb{P}(|\{k \in \mathbb{N} : \mathbb{X} \cap A_k \neq \emptyset\}| = \infty) > 0$. Moreover, by the law of total probability,

$$\mathbb{P}(|\{k \in \mathbb{N} : \mathbb{X} \cap A_k \neq \emptyset\}| = \infty) = \mathbb{E}\Big[\mathbb{P}\Big(|\{k \in \mathbb{N} : \mathbb{X} \cap A_k \neq \emptyset\}| = \infty \Big| \{A_k : k \in \mathbb{N}\}\Big)\Big],$$

and hence (since $\mathbb{X}$ is independent of the random partition $\{A_k : k \in \mathbb{N} \cup \{-1, 0\}\}$), there exists a *deterministic* choice of a partition $\{\hat{A}_k : k \in \mathbb{N} \cup \{-1, 0\}\}$ such that

$$\mathbb{P}\Big(|\{k \in \mathbb{N} \cup \{-1, 0\} : \mathbb{X} \cap \hat{A}_k \neq \emptyset\}| = \infty\Big) > 0,$$

contradicting condition FMV. This completes the proof. ■

3.9.5 Consequences on inductive and self-adaptive learning

Along with optimistically universal *online* learning, [Han21a] identifies two other learning setups, namely *inductive learning* and *self-adaptive learning*.

Inductive learning. An inductive learning rule $\{f_t\}_{t=1}^{\infty}$ is a sequence of measurable functions $f_t : \mathcal{X}^{t-1} \times \mathcal{Y}^{t-1} \times \mathcal{X} \to \mathcal{Y}$ such that given training data $(\mathbb{X}_{<t}, \mathbb{Y}_{<t})$ and input point $X_{t'}$ with $t' > t$ outputs prediction $f_t(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_{t'})$. Its performance is measured in terms of,

$$\mathcal{L}_{\mathbb{X}}(f_t, f^*; t) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t'=t}^{t+T} \ell(f_t(\mathbb{X}_{<t}, \mathbb{Y}_{<t}, X_{t'}), f^*(X_{t'})).$$

Let SUIL denote the set of all processes $\mathbb{X}$ that admit strong universal inductive learning: i.e., for which there exists an inductive learning rule $\{f_t\}$ such that for every measurable $f^* : \mathcal{X} \to \mathcal{Y}$, $\mathcal{L}_{\mathbb{X}}(f_t, f^*; t) \to 0$ $(a.s.)$. Note that the difference between an online learning rule and its inductive counterpart is that the latter will be fixed for an infinite horizon. It was therefore shown by [Han21a] that SUIL $\subset$ SOUL.

Self adaptive learning rule. A self-adaptive learning rule $\{f_{t_1,t_2}\}_{t_1 \leq t_2}^{\infty}$ is a sequence of measurable functions $f_{t_1,t_2} : \mathcal{X}^{t_2-1} \times \mathcal{Y}^{t_1-1} \times \mathcal{X} \to \mathcal{Y}$ such that given training data $(\mathbb{X}_{<t_2}, \mathbb{Y}_{<t_1})$ and input point X_{t_2} it performs prediction $f_{t_1,t_2}(\mathbb{X}_{<t_2}, \mathbb{Y}_{<t_1}, X_{t_2})$. Its performance is measured in terms of

$$\mathcal{L}_{\mathbb{X}}(f_{t_1,\cdot}, f^*; t_1) = \limsup_{T \to \infty} \frac{1}{T} \sum_{t_2=t_1}^{t_1+T} \ell(f_{t_1,t_2}(\mathbb{X}_{<t_2}, \mathbb{Y}_{<t_1}, X_{t_2}), f^*(X_{t_2})).$$

Let SUAL denote the set of all processes $\mathbb{X}$ that admit strong universal self-adaptive learning: i.e., for which there exists a self-adaptive learning rule $\{f_{t_1,t_2}\}$ such that for every measurable $f^* : \mathcal{X} \to \mathcal{Y}$, $\mathcal{L}_{\mathbb{X}}(f_{t_1,\cdot}, f^*; t_1) \to 0 \quad (a.s.)$. Note that self-adaptive learning rules are more expressive than inductive learning rules for they have access to additional unlabeled data, like in the semi-supervised learning setup studied in the literature [CSZ09], yet are still less powerful than online learning rules which would also have access to the respective labels. It was therefore shown by [Han21a] that SUIL $\subset$ SUAL $\subset$ SOUL.

Consequence of Theorem 3.17. For unbounded losses, [Han21a] shows that SUIL = SUAL = SOUL. However, once again this proof relied on the aforementioned complicated arguments. But in light of our proof that SOUL = FS, it becomes immediately apparent that SUIL = SUAL = SOUL and that these classes all admit memorization as an optimistically universal learning rule (merely noting that FS $\subset$ SUIL, since for any $\mathbb{X} \in$ FS, the inductive loss of the memorization rule f_t becomes zero once t exceeds the index of the last novel data point). This greatly simplifies the proof of these equivalences compared to the original proof of [Han21a]. Note that while these three setups turn out to be equivalent when the loss is unbounded, interesting distinctions do exist in the bounded case for which [Han21a] proved that there exists an optimistically universal self-adaptive learning rule (which surprisingly is necessarily different from nearest-neighbor), but no optimistically universal inductive learning rule.

3.10 Conclusion

In this chapter, we characterized universal learning in the realizable setting. Of particular interest, we provided a strong and weak optimistically universal learning rule 2C1NN for bounded losses, which is a simple variant of the nearest neighbor algorithm. We further gave a characterization of the processes admitting strong or weak universal learning.

For bounded losses, a major takeaway is that online learning can be performed well beyond standard statistical assumptions such as stationarity or ergodicity. On the other hand, we saw that the case of unbounded losses is very restrictive. It would be interesting to bridge the gap between these two cases by considering *restricted* universal learning. Specifically, by adding a constraint on the target functions—for example, moment constraints are fairly common in the literature [Gyö+02; GO07]—one could hope to recover the large set of learnable processes SOUL characterized in this chapter, even for the unbounded loss case.

In our setting, we assume that the values are generated from the stochastic process $\mathbb{X}$ through a target function f^* and without noise. Another interesting line of research would

be to add noise to the value process $\mathbb{Y}$. This relates to the Bayes consistency literature in which an objective is to reach the minimal risk, known as the Bayes minimal risk; instead of obtaining exact consistency i.e. vanishing average error rate as considered in this chapter. A possible direction would be to find mild independence conditions on the noise—generalizing the i.i.d. setting [TK22]—so that there exist learning rules which are Bayes universally consistent under a large set of processes $\mathbb{X}$.

Following these two directions, we study universal learning with adversarial noise in the following Chapter 4.

Chapter 4

Universal Regression with Adversarial Responses

4.1 Introduction

We study the classical statistical problem of metric-valued regression. Given an instance metric space $(\mathcal{X}, \rho_{\mathcal{X}})$ and a value metric space $(\mathcal{Y}, \rho_{\mathcal{Y}})$ with a loss ℓ, one observes instances in $\mathcal{X}$ and aims to predict the corresponding values in $\mathcal{Y}$. The learning procedure follows an iterative process where successively, the learner is given an instance X_t and predicts the value Y_t based on the historical samples and the new instance. The learner's goal is to minimize the loss of its predictions $\hat{Y}_t$ compared to the true value Y_t. In particular, $\mathcal{Y} = \{0, 1\}$ (resp. $\mathcal{Y} = \{0, \dots, k\}$) with 0-1 loss corresponds to binary (resp. multiclass) classification while $\mathcal{Y} = \mathbb{R}$ corresponds to the classical regression setting. Motivated by the increase of new types of data in numerous data analysis applications— e.g., data lying on spherical spaces [Cha89; MJM00], manifolds [Shi+09; Dav+10; Fle13], Hilbert spaces [Zai+19], Hadamard spaces [LM21]—we will study the case where both instances and value spaces are general separable metric spaces. This general setting adopted in the literature on universal learning includes and extends the specific classification and regression settings mentioned above. In this context, we model the stream of data as a general stochastic process $(\mathbb{X}, \mathbb{Y}) := (X_t, Y_t)_{t\geq 1}$, and are interested in *consistent* predictions that have vanishing average *excess* loss compared to any fixed measurable predictor functions $f : \mathcal{X} \to \mathcal{Y}$, i.e., $\frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \to 0$ $(a.s.)$. Naturally, one would hope that the algorithm converges for a large class of value functions. Thus, we are interested in *universally consistent* learning rules that are consistent irrespective of the value process $\mathbb{Y}$.

The i.i.d. version of this problem where one assumes that the sequence $(\mathbb{X}, \mathbb{Y})$ is i.i.d. has been extensively studied. A classical result is that for binary classification in Euclidean spaces, $k-$nearest neighbor (kNN) with $k/\ln T \to \infty$ and $k/T \to 0$ is universally consistent under mild assumptions on the distribution of (X_1, Y_1) [Sto77; Dev+94; DGL13]. These results were then extended to a broader class of spaces [DGL13; Gyö+02] and more recently, [Han+21; GW21; CK22] provided universally consistent algorithms for any essentially separable metric space $\mathcal{X}$ which are precisely those for which universal consistency is achievable for i.i.d. pairs $(X_t, Y_t)_{t\geq 1}$ of instances and responses. In parallel, a significant line of work

aimed to obtain such results in non-i.i.d. settings, notably relaxations of the i.i.d. assumptions such as stationary ergodic processes [MYG96; GLM99; Gyö+02] or processes satisfying the law of large numbers [MKN99; GG09; SHS09].

Optimistic universal learning. We aim to understand which are the minimal assumptions on the data sequences for which universal consistency is still achievable. As such, we follow the *optimistic decision theory* [Han21a] which formalizes the paradigm of "learning whenever learning is possible". Precisely, the *provably minimal* assumption for a given objective is that this task is achievable, or in other words that learning is possible. The goal then becomes to 1. characterize for which settings this objective is achievable and 2. if possible, provide learning rules that achieve this objective whenever it is achievable. These are called *optimistically universal* learning rules and enjoy the convenient property that if they failed the objective, any other algorithms would fail as well.

This is precisely the paradigm that we used to sutdy minimal assumptions in the realizable case in Chapter 3. This is the online learning setting in which we assume that there exists an unknown underlying function $f^* : \mathcal{X} \to \mathcal{Y}$ such that $Y_t = f^*(X_t)$. We recall that in this setting, the two questions described above were recently settled. For bounded losses, a simple variant of the nearest neighbor algorithm is optimistically universal and learnable processes are significantly larger than stationary processes. On the other hand, for unbounded losses, universal regression is extremely restrictive since the only learnable processes are those which visit a finite number of points almost surely. In this chapter, we tackle the general non-realizable setting. As an initial result, for bounded losses, [Han22] proposed an algorithm that achieves universal consistency for a large class of processes $\mathbb{X}$, which intuitively asks that the sub-measure induced by empirical visits of the input sequence be continuous (Condition CS). There is however a significant gap between the proposed condition and the learnable processes in the bounded noiseless setting (Condition SMV). [Han22] then left open the question of identifying the precise provably-minimal conditions to achieve consistency, and whether there exists an optimistically universal learning rule.

Adversarial responses and related works in learning with experts. The consistency results in [Han22] hold for arbitrary value processes $\mathbb{Y}$, arbitrarily correlated to the instance process $\mathbb{X}$. We consider the slightly more general *adversarial* responses and show that we can obtain the same results as for adversarial processes, without any generalizability cost. Formally, adversarial responses can not only arbitrarily depend on the instance sequence $\mathbb{X}$, but may also depend on past predictions and past randomness used by the learner. This is a non-trivial generalization for randomized algorithms—note that randomization is necessary to obtain guarantees for general online learning problems [BC+12; Sli+19]. There is a rich theory for arbitrary or adversarial responses $\mathcal{Y}$ when the reference functions $f^* : \mathcal{X} \to \mathcal{Y}$ are restricted to specific function classes $\mathcal{F}$. As a classical example, for the noiseless binary classification setting, there exist learning rules which guarantee a finite number of mistakes for arbitrary sequences $\mathbb{X}$, if and only if the class $\mathcal{F}$ has finite Littlestone dimension [Lit88]. Other restrictions on the function class have been considered [CL06; BPS09; RST15b]. Universal learning diverges from this line of work by imposing no restrictions on function classes—namely *all* measurable functions—but instead restricting instance pro-

cesses $\mathbb{X}$ to the optimistic set where universal consistency is achievable. Nevertheless, the algorithms we introduce for adversarial responses use as subroutine the traditional exponentially weighted forecaster for learning with expert advice from the online learning literature, also known as the Hedge algorithm [LW94; Ces+97b; FS97].

4.1.1 Contributions

In this chapter, we provide answers to two fundamental questions in universal regression. First, we exactly characterize the set of processes we call *learnable*. These are instance processes $\mathbb{X}$ for which universal learning is possible, i.e., consistency is achieved for every process $(X_t, Y_t)_{t\geq 1}$ with covariate sequence $\mathbb{X}$. Second, we provide optimistically universal learning rules, i.e., a unique algorithm that achieves universal consistency for all processes $\mathbb{X}$ for which this is achievable by some learning rule. The specific answers to these questions depend on the value space and loss $(\mathcal{Y}, \ell)$ as detailed below.

Universal learning with empirically integrable responses. We introduce a mild moment-type assumption on the responses $\mathbb{Y}$, namely *empirical integrability*, that roughly asks that one can bound the tails of the empirical first moment of $\mathbb{Y}$. We then proceed to analyze the processes for which learning adversarial responses guaranteed to satisfy this assumption, is achievable. The answer depends on a property of the value space and loss $(\mathcal{Y}, \ell)$ which we denote F-TiME.

- If every ball $B_\ell(y, r)$ of $(\mathcal{Y}, \ell)$ satisfies the F-TiME property, the class of processes $\mathbb{X}$ for which universal consistency under adversarial empirically integrable responses may be achieved is the so-called Sublinear Measurable Visits (SMV) class. This coincides with the class of processes that admits universal learning for bounded losses in the realizable setting (noiseless responses, see Chapter 3). In particular, this shows that for value spaces with bounded losses satisfying F-TiME, one can extend consistency results from the realizable setting to the adversarial one at no generalizability cost.

- Otherwise, the classes of processes $\mathbb{X}$ for which one can achieve universal consistency for empirically integrable responses is a smaller class called Continuous Submeasure (CS). This is a condition that was already considered by [Han22], which showed that for bounded metric losses, one can achieve universal learning under CS processes. Our results show that whenever the F-TiME condition is not satisfied for bounded losses, CS is also a necessary condition for universal learning.

Also, in both cases, we give an optimistically universal learning rule, that is implicit for the first case—it uses as subroutine the learning rule for mean-estimation—and explicit for the second. These results resolve an open question from [Han22].

Intuitively, the property F-TiME asks that, for any fixed tolerance $\epsilon > 0$, there is a learning rule that solves the analogous prediction problem without covariates $\mathbb{X}$—*mean-estimation*—in finite time within the tolerance ϵ. This property is satisfied for "reasonable" value spaces, e.g., totally-bounded spaces or countably-many-classes classification $(\mathbb{N}, \ell_{01})$, but we also provide an explicit example of bounded metric space that does not satisfy this condition.

To motivate the introduction of the empirical integrability condition we show that a weaker moment-type assumption on responses—that $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y_0, Y_t) < \infty$ $(a.s.)$ for some $y_0 \in \mathcal{Y}$—is not sufficient to extend the results from the bounded loss case to unbounded losses, resolving an open question from the universal learning literature. Further, empirical integrability is essentially necessary to obtain consistency results: it is automatically satisfied if the loss is bounded and for the i.i.d. setting it exactly asks that responses Y have finite first moment.

As a direct implication of this work, finite second moment $\mathbb{E}[Y^2]$ is sufficient to achieve consistency for stationary ergodic processes. This result relaxes the conditions of all past works to the best of our knowledge, which required finite fourth moment $\mathbb{E}[Y^4]$ [GO07].

Universal learning with unrestricted responses. For completeness, we also characterize the set of learnable processes without assuming empirical integrability on responses. Since the two notions coincide for bounded losses, we focus on unbounded losses. While there always exists an optimistically universal learning rule, the precise class of universally learnable processes depends on an alternative involving the mean-estimation problem. Either mean-estimation on $(\mathcal{Y}, \ell)$ is impossible and universal learning is never achievable, or universal learning is achievable for processes that only visit a finite number of distinct points, a property called Finite Support (FS). Along the way, we show that mean-estimation with adversarial responses is always possible for metric losses, a result of independent interest.

4.1.2 Organization of the chapter

After presenting the learning framework and definitions in Section 4.2, we describe in Section 4.3 our main results. Although these are stated for general value spaces under the empirical integrability constraint, the proofs build upon the bounded loss case. We follow this proof structure: in Section 4.4 we consider totally-bounded value spaces for which we can give explicit optimistically universal learning rules, in Section 4.5 we consider general bounded loss spaces. We then turn to unbounded and mean estimation in Section 4.6. Last, in Section 4.7 we introduce the empirical integrability and prove our general results for unbounded losses. We conclude discuss open directions in Section 4.8. We give the complete proofs from Sections 4.4 to 4.7 in the appendix Sections 4.9 to 4.12 respectively.

4.2 Formal setup

We use the same framework for universal learning as introduced in Chapter 4. We briefly recall the setup here.

Instance and value spaces. Consider a separable metric *instance space* $(\mathcal{X}, \rho_{\mathcal{X}})$ equipped with its Borel $\sigma-$algebra $\mathcal{B}$, and a separable metric *value space* $(\mathcal{Y}, \rho_{\mathcal{Y}})$ given with a loss ℓ. We recall that a metric space is *separable* if it contains a dense countable set. Unless mentioned otherwise, we suppose that the loss ℓ is a *generalized-metric* on $\mathcal{Y}$ as per Definition 2.1. That is, it is symmetric, positive, and satisfies the following relaxed inequality: for any $0 < \epsilon \leq 1$, there exists a constant c_ϵ^ℓ such that for all $y_1, y_2, y_3 \in \mathcal{Y}$, $\ell(y_1, y_2) \leq (1+\epsilon)\ell(y_1, y_3) + c_\epsilon^\ell \ell(y_2, y_3)$.

An important example for machine learning of generalized metrics are powers of metrics, i.e., there exists $\alpha \geq 1$ such that the loss is $\ell = (\rho_{\mathcal{Y}})^{\alpha}$ (e.g. see [EJ20, Lemma 2.3]). As a remark, all of the results in this work can be generalized to *essentially separable* metric instance spaces, a condition introduced by [Han+21] which was shown to be the largest class of metric spaces for which learning possible. However, for the sake of exposition, we restrict ourselves to separable metric spaces. We denote $\bar{\ell} := \sup_{y_1, y_2 \in \mathcal{Y}} \ell(y_1, y_2)$. In the first Sections 4.4 and 4.5 of this work, we suppose that the loss ℓ is *bounded*, i.e., $\bar{\ell} < \infty$. The case of *unbounded* losses is addressed in the next Sections 4.6 and 4.7. We also recall the notion of near-metrics (Definition 2.1 for which we will provide some results. We say that ℓ is a near-metric on $\mathcal{Y}$ if it is symmetric, positive, and it satisfies a relaxed triangle inequality $\ell(y_1, y_2) \leq c_{\ell}(\ell(y_1, y_3) + \ell(y_2, y_3))$ where c_{ℓ} is a finite constant.

Online learning on adversarial responses. We consider the standard *online learning* framework where at step $t \geq 1$, one observes a new instance $X_t \in \mathcal{X}$ and predicts a value $\hat{Y}_t \in \mathcal{Y}$ based on the past history $(X_u, Y_u)_{u \leq t-1}$ and the new instance X_t only. The learning rule may be randomized, where the private randomness used at each iteration t is drawn from a fixed probability space $\mathcal{R}$ and independent of the data generation process used to generate Y_t. This is the same framework as introduced in Chapter 2, however, we need to carefully explicit the form of the randomness which will be important to define adversarial responses.

Definition 4.1. *An* online learning rule *is a sequence $f. := \{f_t, R_t\}_{t \geq 1}$ of measurable functions $f_t : \mathcal{R} \times \mathcal{X}^{t-1} \times \mathcal{Y}^{t-1} \times \mathcal{X} \to \mathcal{Y}$ together with a distribution R_t on $\mathcal{R}$.*

The prediction at time t of the learning rule $f.$ is $f_t(r_t; (X_u)_{\leq t-1}, (Y_u)_{\leq t-1}, X_t)$ where $r_t \sim R_t$ is independent of the new value X_t and the past history $(X_u, Y_u)_{\leq t}$. For simplicity, we may omit the internal randomness r_t and write directly $f_t : \mathcal{X}^{t-1} \times \mathcal{Y}^{t-1} \times \mathcal{X} \to \mathcal{Y}$. We are interested in general data-generating processes. To this means, a possible very general choice of instances and values are general stochastic processes $(\mathbb{X}, \mathbb{Y}) := \{(X_t, Y_t)\}_{t \geq 1}$ on the product space $\mathcal{X} \times \mathcal{Y}$. This corresponds to the arbitrarily dependent responses under instance processes $\mathbb{X}$ [Han22]. In this chapter, we consider the slightly more general *adversarial responses* where the value Y_t is also allowed to depend on the past private randomness $(r_u)_{u \leq t-1}$ used by the learning rule $f.$.

Definition 4.2. *Let $\mathbb{X} = (X_t)_{t \geq 1}$ be a stochastic process on $\mathcal{X}$. An* adversarial response mechanism *on $\mathbb{X}$ is a stochastic process $\{(\tilde{X}_t, \mathbf{Y}_t)\}_{t \geq 1}$ where $\tilde{X}_t \in \mathcal{X}$, $\mathbf{Y}_t = \mathbf{Y}_t(\cdot \mid \cdot)$ is a Markov kernel from $\mathcal{R}^{t-1}$ to $\mathcal{Y}$, and $(\tilde{X}_t)_{t \geq 1}$ has same distribution as $\mathbb{X}$.*

For a given learning rule $f.$, having observed the sampled randomness $r_1, \ldots, r_{t-1} \in \mathcal{R}$ used by the learning rule before time t, the target value at time t is $Y_t = \mathbf{Y}_t(r_1, \ldots, r_{t-1})$. Again, for simplicity, we will refer to the adversarial response mechanism as $\mathbb{Y}$, which allows us to view the data generating process as a usual stochastic process on $\mathcal{X} \times \mathcal{Y}$. Of course, if the learning rule is *deterministic*, adversarial responses are equivalent to arbitrary dependent responses as in [Han22], but this is not necessarily the case for general *randomized* algorithms.

Empirically integrable responses. We introduce a novel assumption on the responses, namely *empirical integrability*.

Definition 4.3. *A process* $(Y_t)_{t\geq 1}$ *is* empirically integrable *if there exists* $y_0 \in \mathcal{Y}$ *such that for any* $\epsilon > 0$*, almost surely there exists* $M \geq 0$ *for which*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, Y_t) \mathbb{1}_{\ell(y_0,Y_t)\geq M} \leq \epsilon.$$

Unless mentioned otherwise, we will focus on the case where responses satisfy this property. This is a mild assumption on the responses. Indeed, it is worth noting that this condition is always satisfied if the loss ℓ is bounded. Further, if for some $y_0 \in \mathcal{Y}$, $\ell(y_0, Y_t)$ admits moments of order $p > 1$, the empirical integrability condition is also satisfied.

Universal consistency. In this general setting, we are interested in online learning rules which achieve low long-run average loss compared to any fixed prediction function for general adversarial mechanisms. Given a learning rule $f.$ and an adversarial process $(\mathbb{X}, \mathbb{Y})$, for any measurable function $f^* : \mathcal{X} \to \mathcal{Y}$, we denote the long-run average excess loss as

$$\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) := \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\ell(f_t(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t), Y_t) - \ell(f^*(X_t), Y_t)\right).$$

We can then define the notion of consistency which asks that the excess loss compared to any measurable function vanishes to zero.

Definition 4.4. *Let* $(\mathbb{X}, \mathbb{Y})$ *be an adversarial process and* $f.$ *a learning rule.* $f.$ *is consistent under* $(\mathbb{X}, \mathbb{Y})$ *if for any measurable function* $f^* : \mathcal{X} \to \mathcal{Y}$*, we have* $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) \leq 0, \quad (a.s.)$.

For example, if $(\mathbb{X}, \mathbb{Y})$ is an i.i.d. process on $\mathcal{X} \times \mathcal{Y}$ following a distribution μ where μ has a finite first-order moment, achieving consistency is equivalent to reaching the optimal risk $R^* := \inf_{f^*} \mathbb{E}_{(X,Y)\sim\mu} \left[\ell(f^*(X), Y)\right]$, where the infimum is taken over all measurable functions $f^* : \mathcal{X} \to \mathcal{Y}$. As introduced in [Han21a; Han22], consistency against all measurable function is the natural extension of consistency for i.i.d. processes $(\mathbb{X}, \mathbb{Y})$ to non-i.i.d. settings. The goal of universal learning is to design learning rules that are consistent for any adversarial process $\mathbb{Y}$ that is empirically integrable.

Definition 4.5. *Let* $\mathbb{X}$ *be a stochastic process on* $\mathcal{X}$ *and* $f.$ *a learning rule.* $f.$ *is* universally consistent *under* $\mathbb{X}$ *for empirically integrable adversarial responses if for any adversarial process* $(\tilde{\mathbb{X}}, \mathbb{Y})$ *with* $\tilde{\mathbb{X}} \sim \mathbb{X}$ *and such that* $\mathbb{Y}$ *is empirically integrable,* $f.$ *is consistent.*

Optimistic universal learning. Given this regression setup, we define SOLAR (Strong universal Online Learning with Adversarial Responses) as the set of processes $\mathbb{X}$ for which universal consistency with adversarial responses is *achievable*,

$$\text{SOLAR} = \{\mathbb{X} : \exists f. \text{ universally consistent learning rule under } \mathbb{X} \text{ for empirically integrable adversarial responses}\}.$$

Note that this learning rule is allowed to depend on the process $\mathbb{X}$. Similarly, in the realizable (noiseless) setting, one can define the set SOUL (Strong Online Universal Learning) of processes for which there exists a learning rule that is universally consistent for realizable responses when the loss is bounded (and hence, the empirical integrability condition is always satisfied). Of course, SOLAR $\subset$ SOUL. We are then interested in learning rules that would achieve universal consistency whenever possible.

Definition 4.6. *A learning rule f. is* optimistically universal *for adversarial regression with empirically integrable responses if it is universally consistent under all $\mathbb{X} \in$ SOLAR for adversarial empirically integrable responses.*

Similarly, we say that a learning rule is optimistically universal for noiseless regression if it is universally consistent under all $\mathbb{X} \in$ SOUL for noiseless responses when the loss is bounded. In this general framework, the main interests of optimistic learning are 1. identifying the set of learnable processes with adversarial responses SOLAR, 2. determining whether there exists an optimistically universal learning rule, and 3. constructing one if it exists.

4.3 Main results

We introduce some conditions on stochastic processes. For any process $\mathbb{X}$ on $\mathcal{X}$, given any measurable set $A \in \mathcal{B}$ of $\mathcal{X}$, let $\hat{\mu}_{\mathbb{X}}(A) := \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_A(X_t)$. We consider the condition CS (Continuous Sub-measure) which we recall from Condition CS.

Condition CS. *For every decreasing sequence $\{A_k\}_{k=1}^{\infty}$ of measurable sets in $\mathcal{X}$ with $A_k \downarrow \emptyset$, $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_k)] \underset{k\to\infty}{\longrightarrow} 0$.*

It is known that this condition is equivalent to $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(\cdot)]$ being a continuous sub-measure [Han21a], hence the adopted name CS. Importantly, CS processes contain in particular i.i.d., stationary ergodic or stationary processes. We now introduce a second condition SMV (Sublinear Measurable Visits) which asks that for any partition, the process $\mathbb{X}$ visits a sublinear number of sets of the partition. We recall its definition from Condition SMV.

Condition SMV. *For every disjoint sequence $\{A_k\}_{k=1}^{\infty}$ of measurable sets of $\mathcal{X}$ such that $\bigcup_{k=1}^{\infty} A_k = \mathcal{X}$, (every countable measurable partition), $|\{k \geq 1 : A_k \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$, $(a.s.)$.*

This condition is significantly weaker and allows to consider a larger family of processes CS $\subset$ SMV, with CS $\subsetneq$ SMV whenever $\mathcal{X}$ is infinite [Han21a]. Note that these sets depend on the instance space $(\mathcal{X}, \rho_{\mathcal{X}})$. This dependence is omitted for simplicity. We first consider bounded losses. In the *noiseless* case, where there exists some unknown measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ such that the stochastic process $\mathbb{Y}$ is given as $Y_t = f^*(X_t)$ for all $t \geq 1$, we showed in Chapter 3 that learnable processes are exactly SOUL = SMV for bounded losses. We also introduced a learning rule 2-Capped-1-Nearest-Neighbor (2C1NN), variant of the classical 1NN algorithm, which is *optimistically universal* in the noiseless case for bounded losses. Interestingly, we show that this same learning rule is universally consistent for unbounded losses in the noiseless setting with empirically integrable responses.

Theorem 4.1. *Let $(\mathcal{Y}, \ell)$ be a separable near-metric space. Then, 2C1NN is optimistically universal in the noiseless setting with empirically integrable responses, i.e., for all processes $\mathbb{X} \in SMV$ and for all measurable target functions $f^* : \mathcal{X} \to \mathcal{Y}$ such that $(f^*(X_t))_{t\geq 1}$ is empirically integrable, $\mathcal{L}_{(\mathbb{X},(f^*(X_t))_{t\geq 1})}(2C1NN, f^*) = 0$ (a.s.).*

In general, one has SOLAR $\subset$ SMV. It was posed as a question whether we could recover the complete set SMV for learning under adversarial—or arbitrary—processes [Han22].

Question 4.1 ([Han22])**.** *For bounded losses, does there exist an online learning rule that is universally consistent for arbitrary responses under all processes $\mathbb{X} \in SMV (= SOUL)$?*

We answer this question with an alternative. Depending on the bounded value space $(\mathcal{Y}, \ell)$, either SOLAR = SMV or SOLAR = CS, but in both cases there exists an optimistically universal learning rule. We now recall the definition of the Property F-TiME (Finite-Time Mean Estimation) on the value space $(\mathcal{Y}, \ell)$ which characterizes this alternative.

Property F-TiME. *For any $\eta > 0$, there exists a horizon time $T_\eta \geq 1$, an online learning rule $g_{\leq T_\eta}$ such that for any $\boldsymbol{y} := (y_t)_{t=1}^{T_\eta}$ of values in $\mathcal{Y}$ and any value $y \in \mathcal{Y}$, we have*

$$\frac{1}{T_\eta}\mathbb{E}\left[\sum_{t=1}^{T_\eta} \ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)\right] \leq \eta.$$

We are now ready to state our main results for bounded value spaces. The first result shows that if the value space satisfies the above property locally, we can universally learn all the processes in SOUL even under adversarial responses.

Theorem 4.2. *Suppose that any ball of $(\mathcal{Y}, \ell)$, $B_\ell(y, r)$ satisfies F-TiME. Then, SOLAR = SMV and there exists an optimistically universal learning rule $f.$ for adversarial regression with empirically integrable responses., i.e., such that for any stochastic process $(\mathbb{X}, \mathbb{Y})$ on $\mathcal{X} \times \mathcal{Y}$ with $\mathbb{X} \in SMV$ and $\mathbb{Y}$ empirically integrable, for any measurable function $f : \mathcal{X} \to \mathcal{Y}$ we have $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) \leq 0, \quad (a.s.)$.*

F-TiME defines a non-trivial alternative, and an explicit construction of a non-F-TiME bounded metric space $(\mathcal{Y}, \rho_{\mathcal{Y}})$ is given in Section 4.5.1 with $\mathcal{Y} = \mathbb{N}$. Nevertheless, F-TiME is satisfied by a large class of spaces, e.g., any totally-bounded metric space and countable classification $(\mathcal{Y}, \ell) = (\mathbb{N}, \ell_{01})$ satisfy F-TiME. Hence, we can universally learn all SOUL processes with adversarial responses, for countable classification (the empirical integrability condition is automatically satisfied because the loss is bounded). If F-TiME is not satisfied locally, we have the following result which shows that learning under CS is still possible but universal learning beyond CS processes cannot be achieved.

Theorem 4.3. *Suppose that there exists a ball $B_\ell(y, r)$ of $(\mathcal{Y}, \ell)$ that does not satisfy F-TiME. Then, SOLAR = CS and there exists an optimistically universal learning rule $f.$ for adversarial regression with empirically integrable responses., i.e., such that for any stochastic process $(\mathbb{X}, \mathbb{Y})$ on $(\mathcal{X}, \mathcal{Y})$ with $\mathbb{X} \in CS$ and $\mathbb{Y}$ empirically integrable, then, for any measurable function $f : \mathcal{X} \to \mathcal{Y}$ we have $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) \leq 0, \quad (a.s.)$.*

For metric losses $\ell = \rho_{\mathcal{Y}}$, it was already known [Han22] that universal learning under adversarial responses under all processes in CS is achievable by some learning rule. Hence, Theorem 4.3 implies that this learning rule is automatically optimistically universal for adversarial regression for all metric value spaces with bounded loss which do not satisfy F-TiME. However, our result is stronger in that consistency holds for any generalized-metric ℓ, in particular power of a metric losses $\ell = \rho_{\mathcal{Y}}^{\alpha}, \alpha \geq 1$, and for unbounded value spaces.

Remark 4.1. *As a direct consequence of Theorems 4.2 and 4.3, for stationary ergodic processes, finite second moment of the values $\mathbb{E}[Y^2] < \infty$ suffices for consistency, in agreement with the known results for the i.i.d. setting. This relaxes the fourth-moment conditions $\mathbb{E}[Y^4] < \infty$ proposed in the literature [GO07].*

We now consider removing the empirical integrability assumption. As mentioned above, for bounded losses this assumption is automatically satisfied, hence Theorems 4.2 and 4.3 apply directly, with a simplified alternative: whether $(\mathcal{Y}, \ell)$ satisfies F-TiME.

Corollary 4.1. *Suppose that ℓ is bounded.*

- *If $(\mathcal{Y}, \ell)$ satisfies F-TiME. Then, SOLAR = SMV(= SOUL).*
- *If $(\mathcal{Y}, \ell)$ does not satisfy F-TiME. Then, SOLAR = CS.*

Further, an optimistically universal learning rule for adversarial regression always exists, i.e., achieving universal consistency with adversarial responses under any $\mathbb{X} \in$ SOLAR.

It remains to analyze the case of unbounded losses without empirical integrability assumption on the responses. To avoid confusions, we denote by SOLAR-U the set of processes that admit universal learning with adversarial (unrestricted) responses. Unfortunately, even in the noiseless setting, universal learning is extremely restrictive in that case. Specifically, in Chapter 3 we showed that the set of universally learnable processes SOUL for noiseless responses is reduced to the set FS (Finite Support) of processes that visit a finite number of different points almost surely. We briefly recall its formal definition from Condition FS.

Condition FS. *The process $\mathbb{X}$ satisfies $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X} \neq \emptyset\}| < \infty \quad (a.s.)$.*

We show that in the adversarial setting we still have SOLAR-U = FS when ℓ is a metric: we can solve the fundamental problem of mean estimation where one sequentially makes predictions of a sequence $\mathbb{Y}$ of values in $(\mathcal{Y}, \ell)$ and aims to have a better long-run average loss than any fixed value. If responses $\mathbb{Y}$ are i.i.d. this is the Fréchet means estimation problem [EJ20; Sch22; Jaf22; BJ22]. Our main result on mean estimation holds in general spaces and is of independent interest.

Theorem 4.4. *Let $(\mathcal{Y}, \ell)$ be a separable metric space. There exists an online learning rule f. that is universally consistent for adversarial mean estimation, i.e., for any adversarial process $\mathbb{Y}$ on $\mathcal{Y}$, almost surely, for all $y \in \mathcal{Y}$,*

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \left(\ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y, Y_t) \right) \leq 0.$$

Learning setting	Bounded loss	Unbounded loss	Unbounded loss with empirically integrable responses
Noiseless responses	SOUL = SMV [Chapter 3]	SOUL = FS [Chapter 3]	Identical to bounded loss [Chapter 4]
Adversarial (or arbitrary) responses	SOLAR $\supset$ CS (metric loss) [Han22] Does $(\mathcal{Y}, \ell)$ satisfy F-TiME? Yes SOLAR = SMV No SOLAR = CS [Chapter 4]	Is ME achievable? Yes SOLAR-U = FS No SOLAR-U = $\emptyset$ [Chapter 4]	Identical to bounded loss [Chapter 4]

Table 4.1: Characterization of learnable instance processes in universal consistency (ME = Mean Estimation).

Further, we show that for powers of metric we may have SOLAR-U $= \emptyset$. Specifically, for real-valued regression with Euclidean norm and loss $|\cdot|^\alpha$ and $\alpha > 1$, adversarial regression or mean estimation are not achievable. We then show that we have an alternative: either mean estimation with adversarial responses is achievable, SOLAR-U = FS and we have an optimistically universal learning rule; or mean estimation is not achievable and SOLAR-U $= \emptyset$. Thus, even in the best case scenario for unbounded losses, SOLAR-U = FS, which is already extremely restrictive. A natural question is whether imposing moment conditions on the responses would allow recovering the large set SMV as learnable processes instead, which is formalized as follows.

Question 4.2 ([BCH22]). *For unbounded losses ℓ, does there exist an online learning rule $f.$ which is consistent under every $\mathbb{X} \in$ SMV, for every measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ such that there exists $y_0 \in \mathcal{Y}$ with $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f^*(X_t)) < \infty$ (a.s.), i.e., such that we have $\mathcal{L}_{\mathbb{X}}(f., f^*) = 0$ (a.s.)?*

We answer negatively to this question. Under this first-moment condition, universal learning under all SMV processes is not achievable even in this noiseless case. We show the stronger statement that noiseless universal learning under all processes having pointwise convergent relative frequencies—which are included in CS—is not achievable. However, under the empirical integrability condition introduced above we are able to recover all positive results from bounded losses.

Tables 4.1 and 4.2 summarize known results in the literature and our contributions. As a reminder, FS $\subset$ CS $\subset$ SMV in general, and FS $\subsetneq$ CS $\subsetneq$ SMV whenever $\mathcal{X}$ is infinite [Han21a].

[1] In this chapter, an algorithm is optimistically universal if it is universally consistent for all processes under which universal learning is possible in the considered setting. OptiNet, Proto-NN, and MedNet are optimistically universal in another sense, their guarantees hold in all metric spaces for which universal learning with i.i.d. pairs of instances and responses is achievable: *essentially separable* spaces $(\mathcal{X}, \rho_{\mathcal{X}})$ [Han+21]. Our learning rules also enjoy this second optimistic property.

Learning setting	Loss (and response/setting constraints)	Learning rule	Guarantees for which processes $\mathbb{X}$?	Optimist. universal?	Reference
I.i.d. responses	Finite or countable class., 01-loss	OptiNet	i.i.d.	No	[Han+21]
	Real-valued regression + integrable	Proto-NN	i.i.d.	No	[GW21]
	Metric loss + integrable	MedNet	i.i.d.	No	[CK22]
Noiseless responses (realizable)	Bounded loss	2C1NN	SMV	Yes	[Chapter 3]
	Unbounded loss	Memorization	FS	Yes	[Chapter 3]
	Unbounded + EI	2C1NN	SMV	Yes	[Chapter 4]
Adversarial (or arbitrary) responses	Bounded loss + metric loss	Hedge-variant	CS	Not always	[Han22]
	Bounded loss + F-TiME	$(1+\delta)$C1NN-hedged	SMV	Yes	[Chapter 4]
	Bounded loss + not F-TiME	Hedge-variant 2	CS	Yes	[Chapter 4]
	Unbounded loss + ME	ME-algorithm	FS	Yes	[Chapter 4]
	Unbounded loss + not ME	N/A	$\emptyset$	N/A	[Chapter 4]
	Unbounded + EI + local F-TiME	EI-$(1+\delta)$C1NN-hedged	SMV	Yes	[Chapter 4]
	Unbounded + EI + not local F-TiME	EI-Hedge-variant	CS	Yes	[Chapter 4]

Table 4.2: Proposed learning rules for universal consistency (ME = Mean Estimation and EI = Empirical Integrability).[1]

4.4 An optimistically universal learning rule for totally-bounded value spaces

We start our analysis of universal learning under adversarial responses with *totally-bounded* value spaces, for which we can give simple and explicit algorithms. Hence, we suppose in this section that the value space $(\mathcal{Y}, \ell)$ is totally-bounded, i.e., for any $\epsilon > 0$ there exists a finite $\epsilon-$net $\mathcal{Y}_\epsilon$ of $\mathcal{Y}$ such that for any $y \in \mathcal{Y}$, there exists $y' \in \mathcal{Y}_\epsilon$ with $\ell(y, y') < \epsilon$. In particular, a totally-bounded space is necessarily bounded and separable. The goal of this section is to show that for such value spaces, adversarial universal regression is achievable for all processes in SMV as in the noiseless setting (the empirical integrability assumption is automatically satisfied in this context). Further, we explicitly construct an optimistically universal learning rule for adversarial responses.

We recall that in the noiseless setting, the 2C1NN learning rule achieves universal consistency for all SMV processes as shown in Chapter 3. At each iteration t, This rule performs the nearest neighbor rule over a restricted dataset instead of the complete history $\mathbb{X}_{\leq t-1}$. The dataset is updated by keeping track of the number of times each point X_u was used as nearest neighbor. This number is then capped at 2 by deleting from the current dataset any point which has been used twice as representative. Unfortunately, this learning rule is not optimistically universal for adversarial responses. More generally, [CK22] noted that any learning rule which only outputs observed historical values cannot be consistent, even in the simplest case of $\mathcal{X} = \{0\}$ and i.i.d. responses $\mathbb{Y}$. For instance, take $\mathcal{Y} = \bar{B}(0, 1)$ the closed ball of radius 1 in the plane $\mathbb{R}^2$ with the euclidean loss, consider the points $A, B, C \in \mathcal{Y}$ representing the equilateral triangle $e^{2ik\pi/3}$ for $k = 0, 1, 2$, and let $\mathbb{Y}$ be an i.i.d. process following the distribution which visits A, B or C with probability $\frac{1}{3}$. Predictions within observed values, i.e., A, B or C, incur an average loss of $\frac{2}{3}\sqrt{3} > 1$ where 1 is the loss obtained with the fixed value $(0, 0)$.

To construct an optimistically universal learning rule for adversarial responses, we first generalize a result from Chapter 3. Instead of the 2C1NN learning rule, we use $(1+\delta)$C1NN rules for $\delta > 0$ arbitrarily small. Similarly as in 2C1NN, each new input X_t is associated to a

representative $\phi(t)$ used for the prediction $\hat{Y}_t = Y_{\phi(t)}$. In the $(1+\delta)$C1NN rule, each point is used as a representative at most twice with probability δ and at most once with probability $1-\delta$. In order to have this behavior irrespective of the process $\mathbb{X}$, which can be thought of been chosen by a (limited) adversary within the SOUL processes, the information of whether a point can allow for 1 or 2 children is only revealed when necessary. Specifically, at any step $t \geq 1$, the algorithm initiates a search for a representative $\phi(t)$. It successively tries to use the nearest neighbor of X_t within the current dataset and uses it as a representative if allowed by the maximum number of children that this point can have. However, the information whether a potential representative u can have at most 1 or 2 children is revealed only when u already has one child.

- If u allows for 2 children, it will be used as final representative $\phi(t)$.
- Otherwise, u is deleted from the dataset and the search for a representative continues.

The rule is formally described in Algorithm 4.1, where $\bar{y} \in \mathcal{Y}$ is an arbitrary value, and the maximum number of children that a point X_t can have is represented by $1 + U_t$. In this formulation, all Bernoulli $\mathcal{B}(\delta)$ samples are drawn independently of the past history. Note that if $\delta = 1$, the $(1+\delta)$C1NN learning rule coincides with the 2C1NN learning rule.

Theorem 4.5. *Fix $\delta > 0$. For any separable Borel space $(\mathcal{X}, \mathcal{B})$ and any separable near-metric output setting $(\mathcal{Y}, \ell)$ with bounded loss, in the noiseless setting, $(1+\delta)$C1NN is optimistically universal.*

We now construct our algorithm. This learning rule uses a collection of algorithms $f^\epsilon_\cdot$ which each yield an asymptotic error at most a constant factor from $\epsilon^{\frac{1}{\alpha+1}}$. Now fix $\epsilon > 0$ and let $\mathcal{Y}_\epsilon$ be a finite $\epsilon-$net of $\mathcal{Y}$ for ℓ. Recall that we denote by $\bar{\ell}$ the supremum loss. We pose

$$T_\epsilon := \left\lceil \frac{\bar{\ell}^2 \ln |\mathcal{Y}_\epsilon|}{2\epsilon^2} \right\rceil \quad \text{and} \quad \delta_\epsilon := \frac{\epsilon}{2T_\epsilon}.$$

The quantity T_ϵ will be the horizon window used by our learning rule to make its prediction using the $(1+\delta_\epsilon)$C1NN learning rule. Precisely, let ϕ be the representative function from the $(1+\delta_\epsilon)$C1NN learning rule. Note that this representative function $\phi(t)$ is defined only for times t where a new instance X_t is revealed, otherwise $(1+\delta_\epsilon)$C1NN uses simple memorization $\hat{Y}_t = Y_u$. For simplicity, we will denote by $\mathcal{N} = \{t : \forall u < t, X_u \neq X_t\}$ these times of new instances. For $t \in \mathcal{N}$, we denote by $d(t)$ the depth of time t within the graph constructed by $(1+\delta_\epsilon)$C1NN, and define the horizon $L_t = d(t) \mod T_\epsilon$. Intuitively, the learning rule $f^\epsilon_\cdot$ performs the classical Hedge algorithm [CL06] on clusters of times that are close within the graph ϕ. Precisely, we define the equivalence relation between times as follows:

$$t_1 \overset{\phi}{\sim} t_2 \iff \begin{cases} \phi^{L_{u_1}}(u_1) = \phi^{L_{u_2}}(u_2) & \text{and } |\{u < t_i : X_u = X_{t_i}\}| \leq \frac{T_\epsilon}{\epsilon},\ i = 1,2 \\ \text{or} & \\ X_{t_1} = X_{t_2} & \text{and } |\{u < t_i : X_u = X_{t_1}\}| > \frac{T_\epsilon}{\epsilon},\ i = 1,2, \end{cases}$$

where $u_i = \min\{u : X_u = X_{t_i}\}$ is the first occurrence of the considered instance point X_{t_i}. Hence, multiple occurrences of the same instance value fall in the same cluster and for new

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T
Output: Predictions $\hat{Y}_t = (1+\delta)C1NN_t(\boldsymbol{X}_{<t}, \boldsymbol{Y}_{<t}, X_t)$ for $t \leq T$
$\hat{Y}_1 := \bar{y}$ `// Arbitrary prediction at` $t = 1$
$\mathcal{D}_2 \leftarrow \{1\}$; $n_1 \leftarrow 0$; `// Initialisation`
for $t = 2, \ldots, T$ **do**
  if *exists* $u < t$ *such that* $X_u = X_t$ **then**
    $\hat{Y}_t := Y_u$
  else
    $continue \leftarrow True$ `// Begin search for available representative` $\phi(t)$
    while *continue* **do**
      $\phi(t) \leftarrow \min\{l \in \arg\min_{u \in \mathcal{D}_t} \rho_{\mathcal{X}}(X_t, X_u)\}$
      if $n_{\phi(t)} = 0$ **then** `// Candidate representative has no children`
        $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{t\}$
        $continue \leftarrow False$
      else `// Candidate representative has one child`
        $U_{\phi(t)} \sim \mathcal{B}(\delta)$
        if $U_{\phi(t)} = 0$ **then**
          $\mathcal{D}_t \leftarrow \mathcal{D}_t \setminus \{\phi(t)\}$
        else
          $\mathcal{D}_{t+1} \leftarrow (\mathcal{D}_t \setminus \{\phi(t)\}) \cup \{t\}$
          $continue \leftarrow False$
    end
  end
  $\hat{Y}_t := Y_{\phi(t)}$
  $n_{\phi(t)} \leftarrow n_{\phi(t)} + 1$
  $n_t \leftarrow 0$
end

Algorithm 4.1: The $(1+\delta)$C1NN learning rule

instance points times $t \in \mathcal{N}$, all times of a given cluster share the same ancestor up to generation at most $T_\epsilon - 1$. Additionally, a cluster is dedicated to instance points that have a significant number of duplicates. To make its prediction at time t, $f^\epsilon_\cdot$ performs the Hedge algorithm based on values observed on its current cluster $\{u \leq t : u \overset{\phi}{\sim} t\}$. Let $\eta_\epsilon := \sqrt{\frac{8 \ln |\mathcal{Y}_\epsilon|}{\bar{\ell}^2 T_\epsilon}}$ and define the losses $L^t_y = \sum_{u<t : u \overset{\phi}{\sim} t} \ell(Y_u, y)$. The learning rule $f^\epsilon_t(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t)$ outputs a random value in $\mathcal{Y}_\epsilon$ independently from the past history with

$$\mathbb{P}(\hat{Y}_t(\epsilon) = y) = \frac{e^{-\eta_\epsilon L^t_y}}{\sum_{z \in \mathcal{Y}_\epsilon} e^{-\eta_\epsilon L^t_z}}, \quad y \in \mathcal{Y}_\epsilon,$$

where, for simplicity, we denoted $\hat{Y}_t(\epsilon)$ the prediction given by the learning rule $f^\epsilon_\cdot$ at time t.

Having constructed the learning rules $f^\epsilon_\cdot$, we are now ready to define our final learning rule $f_\cdot$. Let $\epsilon_i = 2^{-i}$ for all $i \geq 0$. Intuitively, it aims to select the best prediction within the rules $f^{\epsilon_i}_\cdot$. If there were a finite number of such predictors, we could directly use the algorithms for learning with experts from the literature [CL06]. Instead, we introduce these predictors one

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T,
Representatives $\phi_\epsilon(\cdot)$ and depths $d_\epsilon(\cdot)$ constructed iteratively within $(1+\delta_\epsilon)$C1NN.
Output: Predictions $\hat{Y}_t(\epsilon) = f^\epsilon_t(\boldsymbol{X}_{<t}, \boldsymbol{Y}_{<t}, X_t)$ for $t \le T$
$\mathcal{Y}_\epsilon$ an $\epsilon-$net of $\mathcal{Y}$
$T_\epsilon := \left\lceil \frac{\bar{\ell}^2 \ln |\mathcal{Y}_\epsilon|}{2\epsilon^2} \right\rceil, \quad \eta_\epsilon := \sqrt{\frac{8 \ln |\mathcal{Y}_\epsilon|}{\bar{\ell}^2 T_\epsilon}}$
for $t = 1, \dots, T$ **do**
  $L^t_y = \sum_{u<t : u \overset{\phi_\epsilon}{\sim} t} \ell(Y_u, y), \quad y \in \mathcal{Y}_\epsilon$ `// Losses on the cluster given by` ϕ_ϵ
  $p^t(y) = \dfrac{\exp(-\eta_\epsilon L^t_y)}{\sum_{z \in \mathcal{Y}_\epsilon} \exp(-\eta_\epsilon L^t_z)}, \quad y \in \mathcal{Y}_\epsilon$
  $\hat{Y}_t \sim p^t$
end

Algorithm 4.2: The $f^\epsilon_\cdot$ learning rule

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T,
Predictions $\hat{Y}_\cdot(\epsilon_i)$ from the learning rules $f^{\epsilon_i}_\cdot$.
Output: Predictions $\hat{Y}_t$ for $t \le T$
$w_{0,0} = 1, t_i := \lceil e^i \rceil, \quad i \ge 0$
$I_t = \{i \le \ln t\}, \ \eta_t = \sqrt{\frac{\ln t}{t}}, \quad t \ge 1$
for $t = 1, \dots, T$ **do**
  $L_{t-1,i} := \sum_{s=t_i}^{t-1} \ell(\hat{Y}_s(\epsilon_i), Y_s), \ \hat{L}_{t-1,i} := \sum_{s=t_i}^{t-1} \hat{\ell}_s, \quad i \in I_t$
  $w_{t-1,i} = e^{\eta_t(\hat{L}_{t-1,i} - L_{t-1,i})}$
  $p_t(i) = \dfrac{w_{t-1,i}}{\sum_{j \in I_t} w_{t-1,j}}$
  $\hat{i}_t \sim p_t(\cdot)$ `// model selection`
  $\hat{Y}_t = \hat{Y}_t(\epsilon_i)$
  $\hat{\ell}_t := \frac{\sum_{i \in I_t} w_{t-1,i} \ell(\hat{Y}_t(\epsilon_i), Y_t)}{\sum_{i \in I_t} w_{t-1,i}}$.
end

Algorithm 4.3: An optimistically universal learning rule for totally bounded spaces

at a time: at step $t \ge 1$ we only consider the indices $I_t := \{i \le \ln t\}$. We then compute an estimate $\hat{L}_{t-1,i}$ of the loss incurred by each predictor $f^{\epsilon_i}_\cdot$ for $i \in I_t$ and select a random index $\hat{i}_t$ independent from the past history from an exponentially-weighted distribution based on the estimates $\hat{L}_{t-1,i}$. The final output of our learning rule is $\hat{Y}_t := \hat{Y}_t(\epsilon_{\hat{i}})$. The complete algorithm is formally described in Algorithm 4.3. The following lemma quantifies the loss of the rule $f_\cdot$ compared to the best rule $f^{\epsilon_i}_\cdot$.

Lemma 4.1. *Almost surely, there exists $\hat{t} \ge 0$ such that*

$$\forall t \ge \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^{t} \ell(\hat{Y}_t, Y_t) \le \sum_{s=t_i}^{t} \ell(\hat{Y}_t(\epsilon_i), Y_t) + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{t \ln t}.$$

We are now ready to show that Algorithm 4.3 is universally consistent under SMV processes.

Theorem 4.6. *Suppose that $(\mathcal{Y}, \ell)$ is totally-bounded. There exists an online learning rule $f_\cdot$ which is universally consistent for adversarial responses under any process $\mathbb{X} \in SMV(=$ SOUL$)$, i.e., for any process $(\mathbb{X}, \mathbb{Y})$ on $(\mathcal{X}, \mathcal{Y})$ with adversarial response, such that $\mathbb{X} \in SMV$, then for any measurable function $f : \mathcal{X} \to \mathcal{Y}$, we have $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f_\cdot, f) \leq 0, \quad (a.s.)$.*

Proof sketch. First observe that Lemma 4.1 allows us to combine predictors $f_\cdot^\epsilon$: if individually they perform well, Algorithm 4.3 achieves the best long-term average excess loss among them. We then proceed to show that $f_\cdot^\epsilon$ has low average error in the long run. First, $(1+\delta_\epsilon)$C1NN is universally consistent on SMV processes in the noiseless setting by Theorem 4.5. This intuitively shows that for noiseless functions, the value at time $\phi_\epsilon(t)$ provides a good representative for the value at time t. Extrapolating this argument, we show that if two times are close (for the graph metric) within the graph formed by ϕ_ϵ, they will have close values for any fixed function in the long run. As a result, times in the same cluster defined by $\overset{\phi_\epsilon}{\sim}$ share similar values in the long run. The $f_\cdot^\epsilon$ rule precisely aims to learn the best predictor by cluster using the classical Hedge algorithm. Because it can only ensure low regret compared to a finite number of options, we use ϵ-nets of the value space $\mathcal{Y}$. The reason why we need to have $(1+\delta_\epsilon)$C1NN instead of the known 2C1NN algorithm is that for a given time T, we need to ensure low excess error even though some clusters might not be completed. Because the tree formed by ϕ_ϵ resembles a $(1+\delta_\epsilon)$-branching process, the fraction of times which belong to unfinished clusters is only a small fraction ϵT of the T times, hence does not affect the average long-term excess error significantly. Altogether, we show that $f_\cdot^\epsilon$ has $\mathcal{O}(\epsilon^{\frac{1}{\alpha+1}})$ long-term average excess error compared to any fixed function for any SMV process, which ends the proof.

As a result, SMV $\subset$ SOLAR for totally-bounded value spaces. Recalling that for bounded values SMV = SOUL (Chapter 3), i.e., processes $\mathbb{X} \notin$ SMV are not universally learnable even in the noiseless setting, we have SOLAR $\subset$ SMV. Thus we obtain a complete characterization of the processes which admit universal learning with adversarial responses: SOLAR = SMV. Further, the proposed learning rule is optimistically universal for adversarial regression.

Corollary 4.2. *Suppose that $(\mathcal{Y}, \ell)$ is totally-bounded. Then, SOLAR = SMV, and there exists an optimistically universal learning rule for adversarial regression, i.e., which achieves universal consistency with adversarial responses under any process $\mathbb{X} \in$ SOLAR.*

This is a first step towards the more general Theorem 4.10. Indeed, one can note that F-TiME is satisfied by any totally-bounded value space: given a fixed error tolerance $\eta > 0$, consider a finite $\frac{\eta}{2}-$net $\mathcal{Y}_{\eta/2}$ of $\mathcal{Y}$. Because this is a finite set, we can perform the classical Hedge algorithm [CL06] to have $\Theta(\sqrt{T \ln |\mathcal{Y}_{\eta/2}|})$ regret compared to the best fixed value of $\mathcal{Y}_{\eta/2}$. For example, if $\alpha = 1$, posing $T_\eta = \Theta(\frac{4}{\eta^2} \ln |\mathcal{Y}_{\eta/2}|)$ enables to have a regret at most $\frac{\eta}{2} T_\eta$ compared to any fixed value of $\mathcal{Y}_{\eta/2}$, hence regret at most ηT_η compared to any value of $\mathcal{Y}$. This achieves F-TiME, taking a deterministic time $\tau_\eta := T_\eta$.

4.5 Characterization of learnable processes for bounded losses

While Section 4.4 focused on totally-bounded value spaces, the goal of this section is to give a full characterization of the set SOLAR of processes for which adversarial regression is achievable and provide optimistically universal algorithms, for any *bounded* value space.

4.5.1 Negative result for non-totally-bounded spaces

Although for all bounded value spaces $(\mathcal{Y}, \ell)$, noiseless universal learning is achievable on all SMV(= SOUL) processes, this is not the case for adversarial regression in non-totally-bounded spaces. We show in this section that extending Corollary 4.2 to any bounded value space is impossible: the set of learnable processes for adversarial regression may be reduced to CS only, instead of SMV.

Theorem 4.7. *Let $(\mathcal{X}, \mathcal{B})$ a separable Borel metrizable space. There exists a separable metric value space $(\mathcal{Y}, \ell)$ with bounded loss such that the following holds: for any process $\mathbb{X} \notin CS$, universal learning under $\mathbb{X}$ for arbitrary responses is not achievable. Precisely, for any learning rule $f.$, there exists a process $\mathbb{Y}$ on $\mathcal{Y}$, a measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ and $\epsilon > 0$ such that with non-zero probability $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) \geq \epsilon$.*

In the proof, we explicitly construct a bounded metric space that does not satisfy F-TiME. More precisely, we choose $\mathcal{Y} = \mathbb{N} = \{i \geq 0\}$ and a specific metric loss ℓ with values in $\{0, \frac{1}{2}, 1\}$. For any $k \geq 1$, we pose $n_k := 2k(k-1) + 2^k - 1$ and define the sets

$$I_k := \{n_k, n_k + 1, \dots, n_k + 4k - 1\} \quad \text{and} \quad J_k := \{n_k + 4k, n_k + 4k + 1, \dots, n_{k+1} - 1\}.$$

These sets are constructed so that $|I_k| = 4k$, $|J_k| = 2^k$ for all $k \geq 1$, and together with $\{0\}$, they form a partition of $\mathbb{N}$. We now construct the loss ℓ. We pose $\ell(i,j) = \mathbb{1}_{i=j}$ for all $i, j \in \mathbb{N}$ unless there is $k \geq 1$ such that $(i,j) \in I_k \times J_k$ or $(j,i) \in I_k \times J_k$. It now remains to define the loss $\ell(i,j)$ for all $i \in I_k$ and $j \in J_k$. Note that for any $j \in J_k$, we have that $j - n_k - 4k \in \{0, \dots, 2^k - 1\}$. Hence we will use their binary representation which we write as $j - n_k - 4k = \{b_j^{k-1} \dots b_j^1 b_j^0\}_2 = \sum_{u=0}^{k-1} b_j^u 2^u$ where $b_j^0, b_j^1, \dots, b_j^{k-1} \in \{0,1\}$ are binary digits. Finally, we pose

$$\ell(n_k + 4u, j) = \ell(n_k + 4u + 1, j) = \frac{1 + b_j^u}{2},$$
$$\ell(n_k + 4u + 2, j) = \ell(n_k + 4u + 3, j) = \frac{2 - b_j^u}{2},$$

for all $u \in \{0, 1, \dots, k-1\}$ and $j \in J_k$.

Proof sketch. This value space does not belong to F-TiME because for any algorithm and horizon time k, there is a sequence of length k of elements in I_k with $y_u = n_k + 4(u-1) + 2b_u + c_u$ for $1 \leq u \leq k$ and $b_u, c_u \in \{0,1\}$, such that the algorithm incurs an average excess loss $\frac{1}{4}$ per iteration compared to some fixed element of J_k. To find such a sequence, we sample randomly

and independently Bernoulli variables $b_u, c_u \sim \mathcal{B}(\frac{1}{2})$. In hindsight, the best predictor of the sequence is $n_k + 4k + j$, where $j = b_1 \cdots b_k$ in binary representation. However, the algorithm only observes these bits in an online fashion: at time t it incurs an excess loss cost if it guesses an element of I_k because it has probability at most $\frac{1}{4}$ of finding y_t. And if it predicts an element of J_k, it cannot know in advance the correct t-th bit to choose in their binary representation.

We then proceed to show that for this space SOLAR = CS $\subsetneq$ SOUL. To do so, we show that for processes $\mathbb{X} \notin$ CS there exists a sequence of disjoint measurable sets $\{B_p\}_{p\geq 1}$ and increasing times $(t_p)_{p\geq 1}$ and $\epsilon > 0$ such that with non-zero probability,

$$\forall p \geq 1, \quad \mathbb{X}_{\leq t_{p-1}} \cap B_p = \emptyset \text{ and } \exists t_{p-1} < t \leq t_p : \frac{1}{t}\sum_{t'=1}^{t} \mathbb{1}_{B_p}(X_{t'}) \geq \epsilon.$$

On this event, an online algorithm does not receive any information for instances in B_p before time t_{p-1}. We then construct responses by $(t_{p-1}, t_p]$. During this period and for contexts in B_p, we choose the same difficult-to-predict sequence of values as above for $k = t_p - t_{p-1}$. On the other hand, because the sets B_p are disjoint, there exists a measurable function f^* that selects the best action in hindsight for each set B_p. Intuitively, within horizon t_p, the algorithm cannot gather enough information to achieve lower average excess error than $\frac{\epsilon}{4}$ compared to f^*, which shows that it is not universally consistent.

Although learning beyond CS is impossible in this case, there still exists an optimistically universal learning rule for adversarial responses. Indeed, the main result of [Han22] shows that for any bounded value space, there exists a learning rule which is consistent under all CS processes for arbitrary responses (when ℓ is a metric, i.e., $\alpha = 1$).

Theorem 4.8 ([Han22])**.** *Suppose that $(\mathcal{Y}, \ell)$ is metric and ℓ is bounded. Then, there exists an online learning rule $f.$ which is universally consistent for arbitrary responses under any process $\mathbb{X} \in$ CS, i.e., such that for any stochastic process $(\mathbb{X}, \mathbb{Y})$ on $(\mathcal{X}, \mathcal{Y})$ with $\mathbb{X} \in$ CS, then for any measurable function $f : \mathcal{X} \to \mathcal{Y}$, we have $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f) \leq 0, \quad (a.s.)$.*

The proof of this theorem given in [Han22] extends to adversarial responses. However, we defer the argument because we will later prove Theorem 4.3 which also holds for any generalized-metric loss and unbounded losses in Section 4.7. This shows that for any separable metric space $(\mathcal{X}, \rho_{\mathcal{X}})$, there exists a metric value space for which the learning rule proposed in [Han22] was already optimistically universal.

4.5.2 Adversarial regression for classification with a countable number of classes

Although we showed in the last section that adversarial regression under all SMV processes is not achievable for some non-totally-bounded spaces, we will show that there exist non-totally-bounded value spaces for which we can recover SOLAR = SMV. Precisely, we consider the case of classification with countable number of classes $(\mathbb{N}, \ell_{01})$, with $0-1$ loss $\ell_{01}(i, j) = \mathbb{1}_{i\neq j}$. The goal of this section is to prove that in this case, we can learn arbitrary responses under any SOUL process. The main difficulty with non-totally-bounded classification is that we

cannot apply traditional online learning tools because $\epsilon-$nets may be infinite. Hence, we first show a result that allows us to perform online learning with an infinite number of experts in the context of countable classification.

Lemma 4.2. *Let $t_0 \geq 1$. There exists an online learning rule $f_.$ such that for any sequence $\boldsymbol{y} := (y_i)_{i\geq 1}^T$ of values in $\mathbb{N}$, we have that for $T \geq t_0$*

$$\sum_{t=1}^{T} \mathbb{E}[\ell_{01}(f_t(\boldsymbol{y}_{\leq t-1}), y_t)] \leq \min_{y\in\mathbb{N}} \sum_{t=1}^{T} \ell_{01}(y, y_t) + 1 + \ln 2\sqrt{\frac{t_0}{2\ln t_0}} + \sqrt{\frac{\ln t_0}{2t_0}}(t_0 + T),$$

and with probability $1-\delta$,

$$\sum_{t=1}^{T} \mathbb{E}[\mathbb{1}_{f_t(\boldsymbol{y}_{\leq t-1})=y_t}] \geq \max_{y\in\mathbb{N}} \sum_{t=1}^{T} \mathbb{1}_{y=y_t} - 1 - \ln 2\sqrt{\frac{t_0}{2\ln t_0}} - \sqrt{\frac{\ln t_0}{2t_0}}(t_0 + T) - \sqrt{2T\ln\frac{1}{\delta}}.$$

Proof sketch. We adapt the classical Hedge algorithm, which in its standard form can only ensure sublinear regret compared to a fixed set of values. Instead, we only consider a small subset of candidate values that is enlarged occasionally with previously observed values $y \in \mathbb{Y}_{\leq t}$. This formalizes the intuition that even though there are a priori an infinite number of candidate values ($\mathbb{N}$), it is reasonable to only focus on values with high frequency in the observed sequence $\mathbb{Y}_{\leq t}$: if the next value y_{t+1} is not in this set, the algorithm incurs a loss 1, which would also be incurred by the best fixed predictor until time $t+1$ in hindsight.

We can therefore adapt the learning rules $f_.^\epsilon$ from Section 4.4 by replacing the Hedge algorithm with the algorithm from Lemma 4.2. Further adapting parameters, we obtain our main result for countable classification.

Theorem 4.9. *Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel metrizable space. There exists an online learning rule $f_.$ which is universally consistent for adversarial responses under any process $\mathbb{X} \in$ SMV for countable classification, i.e., such that for any adversarial process $(\mathbb{X}, \mathbb{Y})$ on $(\mathcal{X}, \mathbb{N})$ with $\mathbb{X} \in$ SMV, for any measurable function $f^* : \mathcal{X} \to \mathbb{N}$, we have that $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f_., f^*) \leq 0, \quad (a.s.)$.*

4.5.3 A characterization of universal regression with bounded losses

The last two Sections 4.5.1 and 4.5.2 gave examples of non-totally-bounded value spaces for which we obtain respectively SOLAR = CS or SOLAR = SMV. In this section, we prove that there is an underlying alternative, defined by F-TiME, which enables us to precisely characterize the set SOLAR of learnable processes for adversarial regression.

When F-TiME is satisfied by the value space, similarly to the case of countable classification, we recover SOLAR = SMV and there exists an optimistically universal rule. The corresponding algorithm follows the same general structure as the learning rule provided in Section 4.4 for totally-bounded-spaces, however, the learning rules $f_.^\epsilon$ need to be significantly modified. First, the Hedge algorithm should be replaced by the learning rule $g_{\leq t_\epsilon}$ provided by the F-TiME property. Second, as the horizon time t_ϵ of this learning rule is bounded, the

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T,
Learning rule for finite-time mean estimation $g^{\epsilon}_{\leq t_\epsilon}$, $T_\epsilon = \lceil \frac{t_\epsilon}{\epsilon} \rceil$, $\delta_\epsilon := \frac{\epsilon}{2T_\epsilon}$.
Representatives $\phi_\epsilon(\cdot)$ constructed iteratively within $(1+\delta_\epsilon)$C1NN.
Output: Predictions $\hat{Y}_t(\epsilon) = f^{\epsilon}_t(\boldsymbol{X}_{<t}, \boldsymbol{Y}_{<t}, X_t)$ for $t \leq T$
for $t = 1, \ldots, T$ **do**
  $\mathcal{C}(t) = \{u < t : u \overset{\phi_\epsilon}{\sim} t\}$
  if $\mathcal{C}(t) = \emptyset$ **then** $L_t = 0$ and initialize learner $g^{\epsilon,t}_{\cdot}$;
  else
    $\psi(t) = \max \mathcal{C}(t)$
    if $L_{\psi(t)} < t_\epsilon - 1$ **then** $L_t = L_{\psi(t)} + 1$;
    else $L_t = 0$ and initialize learner $g^{\epsilon,t}_{\cdot}$;
  end
  $\hat{Y}_t = g^{\epsilon,\psi^{L_t}(t)}_{L_t+1}\left(\{y_{\psi^{L_t+1-u}(t)}\}_{u=1}^{L_t}\right)$
end

Algorithm 4.4: The modified $f^{\epsilon}_{\cdot}$ learning rule for value spaces $(\mathcal{Y}, \ell)$ satisfying F-TiME. When initializing a learner $g^{\epsilon,t}_{\cdot}$ for finite-time mean estimation, its internal randomness is sampled independently from the past.

clusters of points on which it is applied have to be adapted: we cannot simply use clusters by distance in the graph defined by the $(1+\delta_\epsilon)C1NN$ algorithm. Instead, we construct clusters of smaller size t_ϵ among these larger graph-based clusters.

More precisely, we take the horizon time t_ϵ and the learning rule $g^{\epsilon}_{\leq t_\epsilon}$ satisfying the condition imposed by the assumption on $(\mathcal{Y}, \ell)$. Then, let $T_\epsilon = \lceil \frac{t_\epsilon}{\epsilon} \rceil$. Similarly as before, we then define $\delta_\epsilon := \frac{\epsilon}{2T_\epsilon}$ and let ϕ be the representative function from the $(1+\delta_\epsilon)$C1NN learning rule. Then, we introduce the same equivalence relation between times $\overset{\phi}{\sim}$, which induces clusters of times. We define a sequence of i.i.d. copies $g^{\epsilon,t}_{\cdot}$ of the learning rule $g^{\epsilon}_{\cdot}$ for all $t \geq 1$. This means that the randomness used within these learning rules is i.i.d, and the copy $g^{\epsilon,t}_{\cdot}$ should be sampled only at time t, independently of the past history. Predictions are then made by blocks of size t_ϵ within the same cluster: at time t, let $u_1 < \ldots < u_{L_t} < t$ be the elements of the current block. If the block does not contain t_ϵ elements yet, we use $g^{\epsilon,u_1}_{L_t+1}$ for the prediction at time t. Otherwise, we start a new block and use $g^{\epsilon,t}_1$. Hence, letting $\psi(t) = \max \mathcal{C}(t)$ be the last time in the same cluster as t (as defined by ϕ_ϵ) and L_t the size of the current block of t without counting t, we now define the learning rule $f^{\epsilon}_{\cdot}$ such that for any sequence $\boldsymbol{x}$, $\boldsymbol{y}$,

$$f^{\epsilon}_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{y}_{\leq t-1}, x_t) := g^{\epsilon,\psi^{L_t}(t)}_{L_t+1}\left(\{y_{\psi^{L_t+1-u}(t)}\}_{u=1}^{L_t}\right).$$

The complete learning rule is given in Algorithm 4.4. The learning rules $f^{\epsilon}_{\cdot}$ are then combined into a single learning rule as in the original algorithm for totally-bounded spaces, following the same procedure given in Algorithm 4.3. We then show that it is universally consistent under SMV processes using same arguments as for Theorem 4.6.

Theorem 4.10. *Suppose that ℓ is bounded and $(\mathcal{Y}, \ell)$ satisfies F-TiME. Then, SOLAR = SMV(= SOUL) and there exists an optimistically universal learning rule for adversarial re-*

gression, i.e., which achieves universal consistency with adversarial responses under any process $\mathbb{X} \in SMV$.

We are now interested in value spaces $(\mathcal{Y}, \ell)$ which do not satisfy F-TiME. We will show that in this case, SOLAR is reduced to the processes CS. We first introduce a second property on value spaces as follows.

Property 2. *For any $\eta > 0$, there exists a horizon time $T_\eta \geq 1$ and an online learning rule $g_{\leq \tau}$ where τ is a random time with $1 \leq \tau \leq T_\eta$ such that for any $\boldsymbol{y} := (y_t)_{t=1}^{T_\eta}$ of values in $\mathcal{Y}$ and any value $y \in \mathcal{Y}$, we have*

$$\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{\tau}\left(\ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)\right)\right] \leq \eta.$$

Remark 4.2. *The random time τ may depend on the possible randomness of the learning rule $g_\cdot$, but it does not depend on any of the values $y_1, y_2, \ldots$ on which the learning rule $g_\cdot$ may be tested. Intuitively, the learning rule uses some randomness which is first privately sampled and may be used by τ. This randomness is never explicitly revealed to the adversary choosing the values $\boldsymbol{y}$, but only implicitly through the realizations of the predictions.*

Lemma 4.3. *Property F-TiME is equivalent to Property 2.*

Using this second property, we can then show that when F-TiME is not satisfied, universal consistency outside CS under adversarial responses is not achievable. In the proof, we only use stochastic processes $(\mathbb{X}, \mathbb{Y})$, hence the same result holds if we only considered universal consistency under arbitrary responses.

Theorem 4.11. *Suppose that ℓ is bounded and $(\mathcal{Y}, \ell)$ does not satisfy F-TiME. Then, SOLAR $=$ CS and there exists an optimistically universal learning rule for adversarial regression, i.e., which achieves universal consistency with adversarial responses under any process $\mathbb{X} \in CS$.*

Proof sketch. First, from Theorem 4.8 we already have CS $\subset$ SOLAR. The main difficulty is to prove that one cannot universally learn any process $\mathbb{X} \notin$ CS. To do so, we re-use the property derived in the proof of Theorem 4.7 that for non-CS processes, one can find a disjoint sequence of sets $\{B_p\}_{p\geq 1}$, an increasing times $(t_p)_{p\geq 1}$ and $\epsilon > 0$ such that with non-zero probability for all $p \geq 1$, the process $\mathbb{X}$ never visits B_p before time t_{p-1} and at some point between times $t_{p-1}+1$ and t_p, the set B_p has been visited a proportion ϵ of times. Now $(\mathcal{Y}, \ell)$ does not satisfy F-TiME, hence does not satisfy Property 2 by Lemma 4.3 for some constant $\eta > 0$. Then, for $p \geq 1$, during period $(t_{p-1}, t_p]$, we define the values $\mathbb{Y}_{t_{p-1}<\cdot\leq t_p}$ when the instance process visits B_p as a sequence $\boldsymbol{y}_{t_{p-1}<\cdot\leq t_p}$ such that the algorithm has average excess loss at least η whenever $\mathbb{X}$ visits B_p, compared to a fixed value $y_p^* \in \mathcal{Y}$. We note that the randomized version of F-TiME given by Lemma 4.3 is important because we do not know in advance when, between t_{p-1} and t_p, B_p has been visited a fraction ϵ of times: potentially, this time is random and there is a huge gap (exponential or more) between t_{p-1} and t_p. On the constructed stochastic process $\mathbb{Y}$, the algorithm does not have vanishing average excess

loss compared to the function equal to y_p^* on B_p. This proves that no algorithm is universally consistent on $\mathbb{X}$.

This completes the proof of Corollary 4.1 and closes our study of universal learning with adversarial responses for bounded value spaces. Notably, there always exists an optimistically universal learning rule, however, this rule highly depends on the value space.

- If $(\mathcal{Y}, \ell)$ satisfies F-TiME, we can learn all SMV = SOUL processes. The proposed learning rule of Theorem 4.10 is *implicit* in general. Indeed, to construct it one first needs to find an online learning rule for mean estimation with finite horizon as described by property F-TiME, which is then used as a subroutine in the optimistically universal learning rule for adversarial regression. We showed however that for totally-bounded value spaces, this learning rule can be *explicited* using $\epsilon-$nets.
- If the value space does not satisfy F-TiME, we can only learn CS processes and there is an inherent gap between noiseless online learning and regression. We propose a learning rule in Section 4.7 which is optimistically universal—see Theorem 4.3. This rule is inspired by the proposed algorithm of [Han22] which is optimistically universal for metric losses $\alpha = 1$.

These two classes of learning rules use very different techniques. Specifically, under processes $\mathbb{X} \in$ CS, [Han21a] showed that there exists a countable set $\mathcal{F}$ of measurable functions $f : \mathcal{X} \to \mathcal{Y}$ which is "dense" within the space of all measurable functions along the realizations $f(X_t)$. We refer to Section 4.7 for a precise description of this density notion. Hence, under process $\mathbb{X}$, we can approximate f^* by functions in $\mathcal{F}$ with arbitrary long-run average precision. However, such property is impossible to obtain for any process $\mathbb{X} \in \text{SMV} \setminus \text{CS}$: no process $\mathbb{X} \notin$ CS admits a "dense" countable sequence of measurable functions. Thus, to learn processes SMV for value spaces satisfying F-TiME, a fundamentally different learning rule than that proposed by [Han21a] or [Han22] was needed.

4.6 Adversarial universal learning for unbounded losses

We now turn to the case of unbounded losses, i.e., value spaces $(\mathcal{Y}, \ell)$ with $\bar{\ell} = \infty$. In this section, we consider universal learning without empirical integrability constraints, for which we introduced the notation SOLAR-U as the set of processes that admit universal learning (we recall that for bounded losses such distinction was unnecessary). In this case, and for more general near-metrics, we showed in Chapter 3 that SOUL = FS. In other terms, for unbounded losses, the learnable processes in the noiseless setting necessarily visit a finite number of distinct instance points of $\mathcal{X}$ almost surely. Thus, universal learning on unbounded value spaces is very restrictive and in particular, SOLAR-U $\subset$ FS. We will show that either SOLAR-U = FS or SOLAR-U = $\emptyset$.

4.6.1 Adversarial regression for metric losses

In this section, we focus on metric losses ℓ, i.e., $\alpha = 1$. In this case, we show that we always have the equality SOLAR-U = FS and that we can provide an optimistically universal

Input: Historical samples $(Y_t)_{t<T}$
Output: Predictions $\hat{Y}_t$ for $t \leq T$
$(y^i)_{i\geq 0}$ dense sequence in $\mathcal{Y}$
$I_t := \{i \leq \ln t : \ell(y^0, y^i) \leq \ln t\}, \eta_t := \frac{1}{4\sqrt{t}}, t \geq 1; \quad t_i = \lceil \max(e^i, e^{\ell(y^0,y^i)}) \rceil, i \geq 0$
$w_{0,0} := 1, \quad \hat{Y}_1 = y^0$ `// Initialisation`
for $t = 2, \ldots, T$ **do**
  $L_{t-1,i} = \sum_{s=t_i}^{t-1} \ell(y^i, Y_s), \quad \hat{L}_{t-1,i} = \sum_{s=t_i}^{t-1} \hat{\ell}_s, \quad i \in I_t$
  $w_{t-1,i} := \exp(\eta_t(\hat{L}_{t-1,i} - L_{t-1,i})), \quad i \in I_t$
  $p_t(i) = \frac{w_{t-1,i}}{\sum_{j\in I_t} w_{t-1,j}}, \quad i \in I_t$
  $\hat{Y}_t \sim p_t(\cdot)$ `// Prediction`
  $\hat{\ell}_t := \frac{\sum_{j\in I_t} w_{t-1,j}\ell(y^j,Y_t)}{\sum_{j\in I_t} w_{t-1,j}}$
end

Algorithm 4.5: The mean estimation algorithm.

learning rule. To do so, we first consider the fundamental estimation problem where one observes values $\mathbb{Y}$ from a general separable metric value space and aims to sequentially predict a value $\hat{Y}_t$ in order to minimize the long-run average loss. We refer to this problem as the mean estimation problem, which is equivalent to regression for the instance space $\mathcal{X} = \{0\}$. For instance, in the specific case of i.i.d. processes $\mathbb{Y}$, mean estimation is exactly the problem of Fréchet mean estimation for distributions on $\mathcal{Y}$. We show that even for adversarial processes $\mathbb{Y}$, we can achieve sublinear regret compared to the best single value prediction, even for unbounded value spaces $(\mathcal{Y}, \ell)$.

If the space were finite, then we could use traditional Hedge algorithms [CL06]. Instead, given a separable value space, we have access to a dense countable sequence of values. We then select the best prediction among this dense sequence by introducing the values of the sequence one at a time, similarly to the argument we used in Lemma 4.1. The learning rule for mean estimation is described in Algorithm 4.5.

Theorem 4.4. *Let $(\mathcal{Y}, \ell)$ be a separable metric space. There exists an online learning rule f. that is universally consistent for adversarial mean estimation, i.e., for any adversarial process $\mathbb{Y}$ on $\mathcal{Y}$, almost surely, for all $y \in \mathcal{Y}$,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (\ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y, Y_t)) \leq 0.$$

Remark 4.3. *The above result guarantees that on the same event of probability one, the proposed learning rule achieves sublinear regret compared to any fixed value prediction. This was not the case for universal regression where, instead, for every fixed measurable function $f : \mathcal{X} \to \mathcal{Y}$, with probability one our learning rules achieved sublinear regret. This stems essentially from the fact that there exists a dense countable set of values $\mathcal{Y}$, but in general, there does not exist a countable set of measurable functions which are dense within all measurable functions in infinity norm.*

We now return to the general regression problem on unbounded spaces. A simple learning rule would be to run in parallel the learning rule g_x for mean estimation on each distinct observed $x \in \mathcal{X}$, i.e., on the sub-process $\mathbb{Y}_{\{t:X_t=x\}}$. As a consequence of Theorem 4.4 we can show that this learning rule is universally consistent on FS processes.

Corollary 4.3. *Suppose that $(\mathcal{Y}, \ell)$ is an unbounded metric space. Then, SOLAR-U $=$ FS$(=$ SOUL$)$ and there exists an optimistically universal learning rule for adversarial regression, i.e., which achieves universal consistency with adversarial responses under any process $\mathbb{X} \in$ FS.*

4.6.2 Negative result for real-valued adversarial regression with loss $\ell = |\cdot|^\alpha$ with $\alpha > 1$

Unfortunately, one cannot extend Corollary 4.3 to losses that are powers of metrics in general. Even in the classical setting of real-valued regression $\mathcal{Y} = \mathbb{R}$ with Euclidean norm, we show that adversarial regression with any loss $\ell = |\cdot|^\alpha$ for $\alpha > 1$ is not achievable, i.e., SOLAR-U $= \emptyset$.

Theorem 4.12. *Let $\alpha > 1$. For the Euclidean value space $(\mathbb{R}, |\cdot|)$ and loss $\ell = |\cdot|^\alpha$ we obtain SOLAR-U $= \emptyset$. In particular, there does not exist a consistent learning rule for mean estimation on $\mathbb{R}$ with squared loss for adversarial responses.*

Proof sketch. The reason why mean estimation with adversarial responses is impossible for $\alpha > 1$ but possible for $\alpha = 1$ is that for $\alpha > 1$, predicting a value off by 1 unit of the best value in hindsight can yield unbounded excess loss for that specific prediction. In particular, we consider a sequence of values of the form $Y_t^{\boldsymbol{b}} = M_t b_t$ where $(M_t)_{t\geq 1}$ is a fixed sequence growing super-exponentially in t, and $\boldsymbol{b} = (b_t)$ is an i.i.d. Rademacher random variables in $\{\pm 1\}$. The sequence $(M_t)_{t\geq 1}$ is constructed so that if the prediction $\hat{Y}_t$ and true value Y_t have different signs $\hat{Y}_t \cdot Y_t \leq 0$, the excess loss of the algorithm compared to the value $sign(Y_t^{\boldsymbol{b}}) = sign(b_t)$ is (super-)linear in t. Because the algorithm cannot know in advance the sign of b_t, there is a realization in which it makes an infinite number of mistakes and as a result has non-zero long-term excess loss compared to the value 1 or -1.

The above of this result also shows that the same negative result holds more generally for unbounded metric value spaces which have some "symmetry". The main ingredients for this negative result were having a point from which there exist arbitrary far values from symmetric directions. In particular, this holds for a discretized value space $(\mathbb{N}, |\cdot|)$ with Euclidean metric, and any Euclidean space $\mathbb{R}^d$ with $d \geq 1$.

4.6.3 An alternative for adversarial regression for unbounded losses

In the two previous sections, we gave examples of losses for which SOLAR-U $= \emptyset$ or we have SOLAR-U $=$ FS. The following simple result is that this is the only alternative and that SOLAR-U $=$ FS is equivalent to achieving consistency for mean estimation with adversarial responses.

Proposition 4.1. *Let $(\mathcal{Y}, \rho_{\mathcal{Y}})$ be a separable metric value space. Suppose that there exists an online learning rule g. which is consistent for mean estimation with adversarial responses for the generalized-metric loss ℓ, i.e., for any adversarial process $\mathbb{Y}$ on $(\mathcal{Y}, \ell)$, we have for any $y^* \in \mathcal{Y}$,*

$$\limsup \frac{1}{T} \sum_{t=1}^{T} (\ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y^*, Y_t)) \leq 0, \quad (a.s),$$

then SOLAR-U $=$ FS *and there exists an optimistically universal learning rule for adversarial regression. Otherwise,* SOLAR-U $= \emptyset$.

Remark 4.4. *There exists separable metric value spaces $(\mathcal{Y}, \rho_{\mathcal{Y}})$ for which powers of metrics losses still yield* SOLAR-U $=$ FS. *For instance, consider $(\mathcal{Y}, \rho_{\mathcal{Y}}) = (\mathbb{R}, \sqrt{|\cdot|_2})$, where $|\cdot|_2$ denotes the Euclidean metric. One can check that this defines a metric on $\mathcal{Y}$ and for any loss $\ell = \rho_{\mathcal{Y}}^{\alpha}$ with $\alpha \leq 2$, we have* SOLAR-U $=$ FS. *However, for $\alpha > 2$,* SOLAR-U $= \emptyset$.

4.7 Adversarial universal online learning with moment constraints

In the previous section, we showed that learnable processes for adversarial regression are only in FS, i.e., visit a finite number of instance points. This shows that universal learning without restrictions on the adversarial responses $\mathbb{Y}$ is extremely restrictive. For instance, it does not contain i.i.d. processes. A natural question is whether adding mild constraints on the process $\mathbb{Y}$ would allow recovering the same results for unbounded losses as for bounded losses from Sections 4.4 and 4.5. This question also arises in noiseless regression since the set of learnable processes is reduced from SOUL $=$ SMV for bounded losses to SOUL $=$ FS for unbounded losses. Hence, a natural question is whether having finite long-run empirical first-order moments would be sufficient to recover learnability in SMV (Question 4.2). Precisely, in this question, the following constraint on noiseless processes $\mathbb{Y} = f^*(\mathbb{X})$ was introduced: there exists $y_0 \in \mathcal{Y}$ with

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f^*(X_t)) < \infty \quad (a.s.).$$

The question now becomes whether there exists an online learning rule which would be consistent under all $\mathbb{X} \in$ SMV processes for any noiseless responses $\mathbb{Y} = f^*(\mathbb{X})$ with f^* satisfying the above first-moment condition. We show that such an objective is not achievable whenever $\mathcal{X}$ is infinite—if $\mathcal{X}$ is finite, any process $\mathbb{X}$ on $\mathcal{X}$ is automatically FS and hence learnable in a noiseless or adversarial setting. In fact, under this first-order moment condition, we show the stronger statement that learning under all processes $\mathbb{X}$ which admit pointwise convergent relative frequencies (CRF) is impossible even in this noiseless setting.

Condition 4.1. *CRF For any measurable set $A \in \mathcal{B}$, $\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_A(X_t)$ exists almost surely.*

[Han21a] showed that CRF $\subset$ CS. In particular, CRF $\subset$ SMV. We show the following negative result on learning under CRF processes for noiseless regression under first-order moment constraint, which holds for unbounded near-metric spaces $(\mathcal{Y}, \ell)$.

Theorem 4.13. *Suppose that $\mathcal{X}$ is infinite and that $(\mathcal{Y}, \ell)$ is an unbounded separable near-metric space. There does not exist an online learning rule which would be consistent under all processes $\mathbb{X} \in$ CRF for all measurable target functions $f^* : \mathcal{X} \to \mathcal{Y}$ such that there exists $y_0 \in \mathcal{Y}$ with*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f^*(X_t)) < \infty \quad (a.s.).$$

Proof sketch. We consider a sequence of values $(y_k)_{k\geq 0}$ such that $\ell(y_0, y_k)$ diverges as $k \to \infty$, then let $(t_k)_{k\geq 1}$ be a sequence of times such that $t_k \approx \sum_{k'\leq k} \ell(y_0, y_k)$. Next, let $(x_k)_{k\geq 0}$ be a sequence of distinct points. We construct a process $\mathbb{X}$ such that $X_t = x_0$ except at sparse times $(t_k)_{k\geq 1}$ for which $X_{t_k} = x_k$. Because t_k has a super-linear growth, $\mathbb{X}$ visits a sublinear number of distinct points and we can show that it satisfies the CRF property. Now for a random binary sequence $\boldsymbol{b} = (b_k)_{kk\geq 1}$ we consider the function $f^*_{\boldsymbol{b}}$ which is equal to y_0 except at points x_k for $k \geq 1$ where $f^*_{\boldsymbol{b}}(x_k) = y_0 \mathbb{1}[b_k = 0] + y_k \mathbb{1}[b_k = 1]$. With these classes of functions, the algorithm cannot know in advance at time t_k whether to predict y_0 or y_k and incurs a loss $\mathcal{O}(\ell(y_0, y_k))$ in average as a result. Therefore, at time t_k, a total loss $\mathcal{O}(\sum_{k'\leq k} \ell(y_0, y_k)) = \mathcal{O}(t_k)$ is incurred compared to $f^*_{\boldsymbol{b}}$. On the other hand, by the construction of the sequence $(t_k)_{k\geq 1}$, $\frac{1}{T}\sum_{t=1}^{T} \ell(y_0, f^*_{\boldsymbol{b}}(X_t)) \leq \frac{1}{T}\sum_{t_k\leq T} \ell(y_0, y_k)$ stays bounded. Thus the learning rule is not consistent under all target functions satisfying the specified moment constraint.

Theorem 4.13 answers negatively to Question 4.2. A natural question is whether another meaningful constraint on responses can be applied to obtain positive results under large classes of processes on $\mathcal{X}$. To this means, we introduced the slightly stronger *empirical integrability* condition. We recall that an (adversarial) process $\mathbb{Y}$ is *empirically integrable* if and only if there exists $y_0 \in \mathcal{Y}$ such that for any $\epsilon > 0$, almost surely there exists $M \geq 0$ with

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, Y_t) \mathbb{1}_{\ell(y_0,Y_t)\geq M} \leq \epsilon.$$

Note that the threshold M may be *dependent* on the adversarial process $\mathbb{Y}$, but the guarantee should hold for any choice of predictions (in the case of adaptive adversaries). This is essentially the mildest condition on the sequence $\mathbb{Y}$ for which we can still obtain results. For example, if the loss is bounded, this constraint is automatically satisfied using $M > \bar{\ell}$. More importantly, note that any process $\mathbb{Y}$ which has bounded higher-than-first moments, i.e., such that there exists $p > 1$ and $y_0 \in \mathcal{Y}$ such that $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell^p(y_0, Y_t) < \infty$, $(a.s.)$, is empirically integrable. Further, for stationary processes $\mathbb{Y}$, having bounded first moment $\mathbb{E}[\ell(y_0, Y_1)] < \infty$ is exactly being empirically integrable. Indeed, by the strong law of large numbers, almost surely $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M} = \mathbb{E}[\ell(y_0, Y_1)\mathbb{1}_{\ell(y_0,Y_1)\geq M}]$. Therefore, empirical integrability is a direct consequence of the dominated convergence theorem.

Lemma 4.4. *Let $\mathbb{Y}$ an stationary process on $\mathcal{Y}$ which has bounded first moment, i.e., there exists $y_0 \in \mathcal{Y}$ such that $\mathbb{E}[\ell(y_0, Y_1)] < \infty$. Then, $\mathbb{Y}$ is empirically integrable.*

Proof Let $\mathbb{Y}$ an stationary process and $y_0 \in \mathcal{Y}$ with $\mathbb{E}[\ell(y_0, Y_1)] < \infty$. Then, by the dominated convergence theorem we have $\mathbb{E}[\ell(y_0, Y_1)\mathbb{1}_{\ell(y_0,Y_1)\geq M}] \to 0$ as $M \to \infty$. Hence, for $\epsilon > 0$, there exists M_ϵ such that $\mathbb{E}[\ell(y_0, Y_1)\mathbb{1}_{\ell(y_0,Y_1)\geq M}] \leq \epsilon$. Then, the sequence $(\ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M})_t$ is still stationary. hence, by the law of large numbers, almost surely,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M_\epsilon} = \mathbb{E}[\ell(y_0, Y_1)\mathbb{1}_{\ell(y_0,Y_1)\geq M_\epsilon}] \leq \epsilon.$$

This ends the proof that $\mathbb{Y}$ is empirically integrable. ■

The goal of this section is to show that under this moment constraint, we can recover all results from Chapter 3, [Han22] and this chapter in Sections 4.4 and 4.5, even for unbounded value spaces, leading up to Theorems 4.2 and 4.3. We will use the following simple equivalent formulation for empirical integrability.

Lemma 4.5. *A process $\mathbb{Y}$ is empirically integrable if and only if there exists $y_0 \in \mathcal{Y}$ such that almost surely, for any $\epsilon > 0$ there exists $M > 0$ with*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M} \leq \epsilon.$$

General strategy. First, the empirical integrability condition holds for some $y_0 \in \mathcal{Y}$ if and only if it holds for all $y_0 \in \mathcal{Y}$. Thus, we can fix $y_0 \in \mathcal{Y}$ independently of the instance or value process. Next, we define the restriction function $\phi_M : \mathcal{Y} \to \mathcal{Y}$ such that $\phi_M(y) = y$ if $\ell(y_0, y) < M$ and $\phi_M(y) = y_0$ otherwise. This function has values in the bounded set $B_\ell(y_0, M)$. Thus, we can apply our learning rules for the bounded loss case to learn the restricted values $\mathbb{Y}^M = (\phi_M(Y_t))_{t\geq 1}$. If we use these predictions to learn $\mathbb{Y}$, the excess loss compared to a fixed function mostly results from the restriction $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(Y_t, \phi_M(Y_t)) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M}$. This excess can then be bounded with the empirical integrability condition at y_0. We then combine the resulting predictors for $M \geq 1$ using Lemma 4.1. While this general strategy allows to use learning rules for the bounded loss case as subroutine to solve the unbounded loss case with empirical integrability constraint, we can adapt it to each case to simplify the algorithms.

4.7.1 Noiseless universal learning with moment condition

We first apply this strategy to the noiseless case. In Chapter 3 we showed that the 2C1NN learning rule achieves universal consistency on all SMV processes for bounded value spaces. Instead of using the 2C1NN learning rule as subroutine as described in the strategy above, we show that we can readily use 2C1NN for empirically integrable noiseless responses in unbounded value spaces, as stated in Theorem 4.1.

To prove this result, first observe that 2C1NN trained on the responses $\mathbb{Y} = (f^*(X_t))_{t\geq 1}$ or the restricted responses $(\phi_M \circ f^*(X_t))_{t\geq 1}$ gives the same prediction at time t provided that

the representative $\phi(t)$ satisfied $\ell(y_0, Y_{\phi(t)}) < M$. By construction of the 2C1NN learning rule, points can be used as representatives at most twice. Hence, up to a factor 2, times when the predictions on unrestricted and restricted responses differ, can be associated with times when $\ell(y_0, Y_t) \geq M$. As a result, we show that the empirical integrability condition can be applied to bound the excess loss resulting from the difference between unrestricted and restricted responses.

4.7.2 Adversarial regression with moment condition under CS processes

We now turn to adversarial regression under CS processes. [Han22] showed that regression for arbitrary responses under all CS processes is achievable in bounded value spaces. We generalize this result to unbounded losses and to adversarial responses with empirical integrability constraint using the general strategy. In particular, our learning rule is also optimistically universal for adversarial regression for all bounded value spaces which do not satisfy F-TiME. Now consider the general case and suppose that there exists a ball $B_\ell(y, r)$ which does not satisfy F-TiME, Theorem 4.11 shows that universal learning for values falling in $B_\ell(y, r)$ cannot be achieved for processes $\mathbb{X} \notin$ CS. Now because $B_\ell(y, r)$ is bounded, responses restricted to this set satisfy the empirical integrability constraint. In particular, this shows that the condition CS is also necessary for universal learning with adversarial responses with empirical integrability. Altogether, this proves Theorem 4.3.

This generalizes the main results from [Han22] to unbounded non-metric losses and from [CK22] to non-metric losses, arbitrary responses and CS instance processes $\mathbb{X}$. Indeed, they consider bounded first moment conditions on i.i.d. responses, which are empirically integrable by Lemma 4.4. Further, as a direct consequence of Theorem 4.3 and Lemma 4.4, we can significantly relax the conditions for universal consistency on stationary ergodic processes found in the literature. Precisely, [GO07] showed that for regression with squared loss, under the assumption $\mathbb{E}[Y_1^4] < \infty$, consistency on stationary ergodic processes is possible. We can relax this result to bounded second moments, matching the standard results for i.i.d. processes.

Corollary 4.4. *Let $(\mathcal{Y}, \ell) = (\mathbb{R}, |\cdot|^2)$. The learning rule of Theorem 4.3 is consistent on any stationary ergodic process $(X_t, Y_t)_{t\geq 1}$ with $\mathbb{E}[Y_1^2] < \infty$.*

4.7.3 Adversarial regression with moment condition under SMV processes

Last, we generalize our result Theorem 4.10 for value spaces satisfying F-TiME, to unbounded value spaces, with the same moment condition on responses using the general strategy. In order to apply Theorem 4.10 to bounded balls of the value space, we now ask that all balls $B_\ell(y, r)$ in the value space $(\mathcal{Y}, \ell)$ satisfy F-TiME. This proves Theorem 4.2.

Theorems 4.2 and 4.3 completely characterize learnability for adversarial regression with moment condition. Namely, if the value space $(\mathcal{Y}, \ell)$ is such that any bounded ball satisfies F-TiME (resp. there exists a ball $B_\ell(y, r)$ that disproves F-TiME), Theorem 4.2 (resp. Theorem 4.3) gives an optimistic learning rule which achieves consistency under all processes

in SMV (resp. CS). This ends our analysis of adversarial regression for unbounded value spaces.

4.8 Conclusion

In this work, we provided a characterization of learnability for universal learning in the regression setting, for a class of generalized-metric losses satisfying specific relaxed triangle inequality identities, which contains powers of metrics $\ell = \rho_{\mathcal{Y}}^{\alpha}$ for $\alpha \geq 1$. A natural question would be whether one can generalize these results to larger classes of losses, e.g. non-symmetric losses which may appear in classical machine learning problems.

The present work also has some implications for adversarial contextual bandits. Specifically, one may consider the case of a learner who receives partial information on the rewards/losses as opposed to the traditional regression setting where the response is completely revealed at each iteration. In the latter case, the learner can for instance compute the loss of *all* values with respect to the response realization. On the other hand, in the contextual bandits framework, the reward/loss is revealed *only* for the pulled arm—or equivalently the prediction of the learner. In these partial information settings, exploration then becomes necessary. This is precisely the direction that we pursue in Chapters 5 and 6.

4.9 Appendix A: Proofs of Section 4.4

4.9.1 Proof of Theorem 4.5

In this section, we prove that for any $\delta > 0$, the $(1+\delta)$C1NN learning rule is optimistically universal for the noiseless setting. The proof follows the same structure as the proof of the result Chapter 3 which shows that 2C1NN is optimistically universal. We first focus on the binary classification setting and show that the learning rule $(1+\delta)$C1NN is consistent on functions representing open balls.

Proposition 4.2. *Fix $0 < \delta \leq 1$. Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space constructed from the metric $\rho_{\mathcal{X}}$. We consider the binary classification setting $\mathcal{Y} = \{0,1\}$ and the ℓ_{01} binary loss. For any input process $\mathbb{X} \in SMV$, for any $x \in \mathcal{X}$, and $r > 0$, the learning rule $(1+\delta)C1NN$ is consistent for the target function $f^* = \mathbb{1}_{B_{\rho_{\mathcal{X}}}(x,r)}$.*

Proof We fix $\bar{x} \in \mathcal{X}$, $r > 0$ and $f^* = \mathbb{1}_{B(\bar{x},r)}$. We reason by the contrapositive and suppose that $(1+\delta)$C1NN is not consistent on f^*. Then, $\eta := \mathbb{P}(\mathcal{L}_{\mathbb{X}}((1+\delta)C1NN, f^*) > 0) > 0$. Therefore, there exists $0 < \epsilon \leq 1$ such that $\mathbb{P}(\mathcal{L}_{\mathbb{X}}((1+\delta)C1NN, f^*) > \epsilon) > \frac{\eta}{2}$. Denote by $\mathcal{A} := \{\mathcal{L}_{\mathbb{X}}((1+\delta)C1NN, f^*) > \epsilon\}$. this event of probability at least $\frac{\eta}{2}$. Because $\mathcal{X}$ is separable, let $(x^i)_{i\geq 1}$ a dense sequence of $\mathcal{X}$. We consider the same partition $(P_i)_{i\geq 1}$ of $B(\bar{x}, r)$ and the partition $(A_i)_{i\geq 0}$ of $\mathcal{X}$ as in the original proof in Chapter 3 (Proposition 3.3), but with the constant $c_\epsilon := \frac{1}{2\cdot 2^{2^8/(\epsilon\delta)}}$ and changing the construction of the sequence $(n_l)_{l\geq 1}$ so that for all $l \geq 1$

$$\mathbb{P}\left[\forall n \geq n_l,\ |\{i,\ P_i(\tau_l) \cap \mathbb{X}_{<n} \neq \emptyset\}| \leq \frac{\epsilon\delta}{2^{10}} n\right] \geq 1 - \frac{\delta}{2\cdot 2^{l+2}} \quad \text{and} \quad n_{l+1} \geq \frac{2^9}{\epsilon\delta} n_l.$$

Last, consider the product partition of $(P_i)_{i\geq 1}$ and $(A_i)_{i\geq 0}$ which we denote $\mathcal{Q}$. Similarly, we define the same events $\mathcal{E}_l, \mathcal{F}_l$ for $l \geq 1$. We aim to show that with nonzero probability, $\mathbb{X}$ does not visit a sublinear number of sets of $\mathcal{Q}$.

We now denote by $(t_k)_{k\geq 1}$ the increasing sequence of all (random) times when $(1+\delta)$C1NN makes an error in the prediction of $f^*(X_t)$. Because the event $\mathcal{A}$ is satisfied, $\mathcal{L}_{\boldsymbol{x}}((1+\delta)C1NN, f^*) > \epsilon$, we can construct an increasing sequence of indices $(k_l)_{l\geq 1}$ such that $t_{k_l} < \frac{2k_l}{\epsilon}$. For any $t \geq 2$, we will denote by $\phi(t)$ the (random) index of the representative chosen by the $(1+\delta)$C1NN learning rule. Now let $l \geq 1$. Consider the tree $\mathcal{G}$ where nodes are times $\mathcal{T} := \{t \leq t_{k_l}\}$ within horizon t_{k_l}, where the parent relations are given by $(t, \phi(t))$ for $t \in \mathcal{T} \setminus \{1\}$. In other words, we construct the tree in which the parent of each new input is its representative. Note that by construction of the $(1+\delta)$C1NN learning rule, each node has at most 2 children.

Step 1

In this step, we consider the case when the majority of input points on which $(1+\delta)$C1NN made a mistake belong to $B(\bar{x}, r)$, i.e., $|\{k \leq k_l,\ X_{t_k} \in B(\bar{x}, r)\}| \geq \frac{k_l}{2}$. We denote $\mathcal{H}_1$ this event. Let us now consider the subgraph $\tilde{\mathcal{G}}$ given by restricting $\mathcal{G}$ only to nodes in the ball $B(\bar{x}, r)$—which are mapped to the true value 1—i.e., on times $\mathcal{T} := \{t \leq t_{k_l},\ X_t \in B(\bar{x}, r)\}$. In this subgraph, the only times with no parent are times t_k with $k \leq k_l$ and $X_{t_k} \in B(\bar{x}, r)$, and possibly time $t = 1$. Therefore, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with roots times $\{t_k,\ k \leq k_l,\ x_{t_k} \in B(\bar{x}, r)\}$, and possibly $t = 1$ if $X_1 \in B(\bar{x}, r)$. For a given time t_k with $k \leq k_l$ and $X_{t_k} \in B(\bar{x}, r)$, we denote by $\mathcal{T}_k$ the corresponding tree in $\tilde{\mathcal{G}}$ with root t_k. We now introduce the notion of *good* trees. We say that $\mathcal{T}_k$ is a good tree if $\mathcal{T}_k \cap \mathcal{D}_{t_{k_l}+1} \neq \emptyset$, i.e., the tree survived until the last dataset. Conversely a tree is *bad* if all its nodes were deleted before time $t_{k_l} + 1$. We denote the set of good and bad trees by $G = \{k : \mathcal{T}_k \text{ good}\}$ and $B = \{k : \mathcal{T}_k \text{ bad}\}$. In particular, we have $|G| + |B| = |\{k \leq k_l, X_{t_k} \in B(\bar{x}, r)\}| \geq k_l/2$. We aim to upper bound the number of bad trees. We now focus on trees $\mathcal{T}_k$ which induced a future first mistake, i.e., such that $\{l \in \mathcal{T}_k | \exists u \leq t_{k_l} : \phi(u) = l, \rho_{\mathcal{X}}(X_l, \bar{x}) \geq r \text{ and } \forall v < u, \phi(v) \neq l\} \neq \emptyset$. We denote the corresponding minimum time $l_k = \min\{l \in \mathcal{T}_k \mid \exists u \leq t_{k_l} : \phi(u) = l, \rho_{\mathcal{X}}(X_l, \bar{x}) \geq r, \forall v < u, \phi(v) \neq l\}$. The terminology first mistake refers to the fact that the first time which used l as representative corresponded to a mistake, as opposed to l already having a children $X_u \in B(\bar{x}, r)$ which continues descendents of l within the tree $\mathcal{T}_k$. Note that bad trees necessarily induce a future first mistake—otherwise, this tree would survive. For each of these times l_k two scenarios are possible.

1. The value U_{l_k} was never revealed within horizon t_{k_l}: as a result $l_k \in \mathcal{D}_{t_{k_l}+1}$.
2. The value U_{l_k} was revealed within horizon t_{k_l}. Then, U_{l_k} we revealed using a time t for which l_k was a potential representative. This scenario has two cases:
   (a) $\rho_{\mathcal{X}}(X_t, \bar{x}) < r$. If used as representative $\phi(t) = l_k$, then l_k would not have induced a mistake in the prediction of Y_t.
   (b) $\rho_{\mathcal{X}}(X_t, \bar{x}) \geq r$. If used as representative $\phi(t) = l_k$, then l_k would have induced a mistake in the prediction of Y_t.

In the case 2.a), if the point is used as representative $\phi(t) = l_k$ and if the corresponding tree $\mathcal{T}_k$ was bad, at least another future mistake is induced by $\mathcal{T}_k$—otherwise this tree would survive. We consider times l_k for which the value was revealed, which corresponds to the only possible scenario for bad trees. We denote the corresponding set $K := \{k : U_{l_k} \text{ revealed within horizon } t_{k_l}\}$. We now consider the sequence $k_1^a, \dots k_\alpha^a$ containing all indices of K for which scenario 2.a) was followed, ordered by chronological order for the reveal of $U_{l_{k_i^a}}$, i.e., $U_{l_{k_1^a}}$ was the first item of scenario 2.a) to be revealed, then $U_{l_{k_2^a}}$ etc. until $U_{l_{k_\alpha^a}}$. Similarly, we construct the sequence $k_1^b, \dots k_\beta^b$ of indices in K corresponding to scenario 2.b), ordered by order for the reveal of $U_{l_{k_i^b}}$. We now consider the events

$$\mathcal{B} := \left\{\alpha + \beta \leq \frac{k_l}{2} - \frac{k_l\delta}{32}\right\}, \quad \mathcal{C} := \left\{\sum_{i=1}^{\min(\alpha, \lceil k_l/8\rceil)} U_{l_{k_i^a}} \geq \frac{k_l\delta}{16}\right\},$$
$$\mathcal{D} := \left\{\sum_{i=1}^{\min(\beta, \lceil k_l/8\rceil)} U_{l_{k_i^b}} \geq \frac{k_l\delta}{16}\right\}.$$

We now show that for $l > 16$, under the event

$$\mathcal{M}_{k_l} := \mathcal{H}_1 \cap \left[\mathcal{B} \cup \left(\{\alpha \geq \lceil k_l/8\rceil\} \cap \mathcal{C}\right) \cup \left(\{\alpha < \lceil k_l/8\rceil\} \cap \mathcal{D}\right)\right],$$

we have that $|G| \geq \frac{k_l\delta}{32}$. Suppose that $\mathcal{M}_{k_l}$ is met. First note that because a bad tree can only fall into scenarios 2.a) or 2.b) we have $|B| \leq \alpha + \beta$. Hence $|G| \geq \frac{k_l}{2} - \alpha - \beta$ because of $\mathcal{H}_1$. Thus, the result holds directly if $\mathcal{B}$ is satisfied. We can now suppose that $\mathcal{B}^c$ is satisfied, i.e., $\alpha + \beta > \frac{k_l}{2} - \frac{k_l\delta}{32}$. Now suppose that $\alpha \geq \lceil k_l/8\rceil$ and $\mathcal{C}$ are also satisfied. For all indices such that $U_{l_{k_i^a}} = 1$, i.e., we fall in case 2.a) and $l_{k_i^a}$ is used as representative, the corresponding tree $\mathcal{T}_{k_i^a}$ would need to induce at least an additional mistake to be bad. Recall that in total at most $k_l/2$ mistakes are induced by points of $\mathcal{T}$. Also, by definition of the set K, $\alpha + \beta$ mistakes are already induced by the times t_k for $k \in K$. These corresponded to the future first mistakes for all times $\{l_k : k \in K\}$. Hence, we obtain

$$|G| \geq \sum_{i=1}^{\alpha} U_{l_{k_i^a}} - \left(\frac{k_l}{2} - \alpha - \beta\right) \geq \frac{k_l\delta}{16} - \frac{k_l\delta}{32} = \frac{k_l\delta}{32}.$$

Now consider the case where $\mathcal{H}_1$, $\mathcal{B}^c$, $\alpha < \lceil k_l/8\rceil$ and $\mathcal{D}$ are met. In particular, because $l > 16$ we have $k_l > 16$ hence $\frac{k_l}{2} - \frac{k_l\delta}{32} \geq 2\lceil k_l/8\rceil$. Thus, because of $\mathcal{B}^c$ we have $\beta > \frac{k_l}{2} - \frac{k_l\delta}{32} - \alpha \geq \lceil k_l/8\rceil$. Now observe that for all indices such that $U_{l_{k_i^b}} = 1$, the time l_k induced two mistakes. Therefore, counting the total number of mistakes we obtain

$$\frac{k_l}{2} \geq \alpha + \beta + \sum_{i=1}^{\beta} U_{l_{k_i^b}} \geq \frac{k_l}{2} - \frac{k_l\delta}{32} + \frac{k_l\delta}{16}$$

which is impossible. This ends the proof that under $\mathcal{M}_{k_l}$ we have $|G| \geq \frac{k_l\delta}{32}$.

We now aim to lower bound the probability of this event. To do so, we first upper bound the probability of the event $\{\alpha \geq \lceil k_l/8\rceil\} \cap \mathcal{C}^c$. We introduce a process $(Z_i)_{i=1}^{\lceil k_l/8\rceil}$ such that

for all $i \leq \max(\alpha, \lceil k_l/8 \rceil)$, $Z_i = U_{l_{k_i^a}} - \delta$ and $Z_i = 0$ for $\alpha < i \leq \lceil k_l/8 \rceil$. Because of the specific ordering chosen $k_1^a, \ldots, k_\alpha^a$, this process is a sequence of martingale differences, with values bounded by 1 in absolute value. Therefore, for $l > 16$ the Azuma-Hoeffing inequality yields

$$\mathbb{P}\left[\sum_{i=1}^{\lceil k_l/8 \rceil} Z_i \leq -\frac{k_l \delta}{16}\right] \leq e^{-\frac{k_l^2 \delta^2}{2 \cdot 16^2 (k_l/8+1)}} \leq e^{-\frac{k_l \delta^2}{2^7}}.$$

But on the event $\{\alpha \geq \lceil k_l/8 \rceil\} \cap \mathcal{C}^c$ we have precisely

$$\sum_{i=1}^{\lceil k_l/8 \rceil} Z_i = \sum_{i=1}^{\min(\alpha, \lceil k_l/8 \rceil)} U_{l_{k_i^a}} - \lceil k_l/8 \rceil \delta \leq \frac{k_l \delta}{16} - \lceil k_l/8 \rceil \delta \leq -\frac{k_l \delta}{16}.$$

Therefore $\mathbb{P}[\mathcal{C}^c \cap \{\alpha \geq \lceil k_l/8 \rceil\}] \leq \mathbb{P}\left[\sum_{i=1}^{\lceil k_l/8 \rceil} Z_i \leq -\frac{k_l \delta}{16}\right] \leq e^{-k_l \delta^2/2^7}$. Similarly we obtain $\mathbb{P}[D^c \cap \{\beta \geq \lceil k_l/8 \rceil\}] \leq e^{-k_l \delta^2/2^7}$. Finally we write for any $l > 16$,

$$\begin{aligned}
\mathbb{P}[\mathcal{H}_1 \setminus \mathcal{M}_{k_l}] &= \mathbb{P}[\mathcal{H}_1 \cap \mathcal{B}^c \cap (\{\alpha < \lceil k_l/8 \rceil\} \cup \mathcal{C}^c) \cap (\{\alpha \geq \lceil k_l/8 \rceil\} \cup \mathcal{D}^c)] \\
&= \mathbb{P}[\mathcal{H}_1 \cap \mathcal{B}^c \cap [(\{\alpha < \lceil k_l/8 \rceil\} \cap \mathcal{D}^c) \cup (\{\alpha \geq \lceil k_l/8 \rceil\} \cap \mathcal{C}^c)]] \\
&\leq \mathbb{P}[\mathcal{C}^c \cap \{\alpha \geq \lceil k_l/8 \rceil\}] + \mathbb{P}[\mathcal{D}^c \cap \{\alpha < \lceil k_l/8 \rceil\} \cap \mathcal{B}^c] \\
&\leq \mathbb{P}[\mathcal{C}^c \cap \{\alpha \geq \lceil k_l/8 \rceil\}] + \mathbb{P}[\mathcal{D}^c \cap \{\beta \geq \lceil k_l/8 \rceil\}] \\
&\leq 2e^{-\frac{k_l \delta^2}{2^7}}.
\end{aligned}$$

In particular, we obtain

$$\mathbb{P}\left[\left\{|G| \geq \frac{k_l \delta}{32}\right\} \cap \mathcal{H}_1\right] \geq \mathbb{P}[\mathcal{M}_{k_l}] \geq \mathbb{P}[\mathcal{H}_1] - 2e^{-\frac{k_l \delta^2}{2^7}}.$$

Step 2

We now consider the opposite case, when a majority of mistakes are made outside $B(\bar{x}, r)$, i.e., $|\{k \leq k_l,\ X_{t_k} \in B(\bar{x}, r)\}| < \frac{k_l}{2}$, which corresponds to the event $\mathcal{H}_1^c$. Similarly, we consider the subgraph $\tilde{\mathcal{G}}$ given by restricting $\mathcal{G}$ only to nodes outside the ball $B(\bar{x}, r)$, i.e., on times $\mathcal{T} := \{t \leq t_{k_l},\ \rho_{\mathcal{X}}(X_t, \bar{x}) \geq r)\}$. Again, $\tilde{\mathcal{G}}$ is a collection of disjoint trees with roots times $\{t_k,\ k \leq k_l,\ \rho_{\mathcal{X}}(X_{t_k}, \bar{x}) \geq r)\}$—and possibly $t = 1$. For a given time t_k with $k \leq k_l$ and $\rho_{\mathcal{X}}(X_{t_k}, \bar{x}) \geq r$, we denote by $\mathcal{T}_k$ the corresponding tree in $\tilde{\mathcal{G}}$ with root t_k. Similarly to the previous case, $\mathcal{T}_k$ is a *good* tree if $\mathcal{T}_k \cap \mathcal{D}_{t_{k_l}+1} \neq \emptyset$ and *bad* otherwise. We denote the set of good and bad trees by $G = \{k : \mathcal{T}_k \text{ good}\}$. We can again focus on trees $\mathcal{T}_k$ which induced a future first mistake, i.e., such that $\{l \in \mathcal{T}_k | \exists u \leq t_{k_l} : \phi(u) = l, \rho_{\mathcal{X}}(X_l, \bar{x}) < r \text{ and } \forall v < u, \phi(v) \neq l\} \neq \emptyset$ and more specifically their minimum time $l_k = \min\{l \in \mathcal{T}_k \mid \exists u \leq t_{k_l} : \phi(u) = l, \rho_{\mathcal{X}}(X_l, \bar{x}) < r, \forall v < u, \phi(v) \neq l\}$. The same analysis as above shows that

$$\mathbb{P}\left[\left\{|G| \geq \frac{k_l \delta}{32}\right\} \cap \mathcal{H}_1^c\right] \geq \mathbb{P}[\mathcal{H}_1^c] - 2e^{-\frac{k_l \delta^2}{2^7}}.$$

Therefore, if G denotes more generally the set of good trees (where we follow the corresponding case 1 or 2) we finally obtain that for any $l > 16$,

$$\mathbb{P}\left[|G| \geq \frac{k_l \delta}{32}\right] \geq 1 - 4e^{-\frac{k_l \delta^2}{2^7}}.$$

We denote by $\tilde{\mathcal{M}}_{k_l}$ this event. By Borel-Cantelli lemma, almost surely, there exists $\hat{l}$ such that for any $l \geq \hat{l}$, the event $\tilde{\mathcal{M}}_{k_l}$ is satisfied. We denote $\mathcal{M} := \bigcup_{l \geq 1} \bigcap_{l' \geq l} \tilde{\mathcal{M}}_{k_l}$ this event of probability one. The aim is to show that on the event $\mathcal{A} \cap \mathcal{M} \cap \bigcap_{l \geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)$, which has probability at least $\frac{\eta}{4}$, $\mathbb{X}$ disproves the SMV condition. In the following, we consider a specific realization $\boldsymbol{x}$ of the process $\mathbb{X}$ falling in the event $\mathcal{A} \cap \mathcal{M} \cap \bigcap_{l \geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)$—$\boldsymbol{x}$ is not random anymore. Let $\hat{l}$ be the index given by the event $\mathcal{M}$ such that for any $l \geq \hat{l}$, $\mathcal{M}_{k_l}$ holds. We consider $l \geq \hat{l}$ and successively consider different cases in which the realization $\boldsymbol{x}$ may fall.

- In the first case, we suppose that a majority of mistakes were made in $B(\bar{x}, r)$, i.e., that we fell into event $\mathcal{H}_1$ similarly to Step 1. Because the event $\tilde{\mathcal{M}}_{k_l}$ is satisfied we have $|G| \geq \frac{k_l \delta}{2^5}$. Now note that trees are disjoint, therefore, $\sum_{k \in G} |\mathcal{T}_k| \leq t_{k_l} < \frac{2k_l}{\epsilon}$. Therefore,

$$\sum_{k \in G} \mathbb{1}_{|\mathcal{T}_k| \leq \frac{2^7}{\epsilon\delta}} = |G| - \sum_{k \in G} \mathbb{1}_{|\mathcal{T}_k| > \frac{2^7}{\epsilon\delta}} > |G| - \frac{\epsilon\delta}{2^7} \sum_{k \in G} |\mathcal{T}_k| \geq \frac{k_l\delta}{2^5} - \frac{k_l\delta}{2^6} = \frac{k_l\delta}{2^6}.$$

  We will say that a tree $|\mathcal{T}_k|$ is *sparse* if it is good and has at most $\frac{2^7}{\epsilon\delta}$ nodes. With $S := \{k \in G,\ |\mathcal{T}_k| \leq \frac{2^7}{\epsilon\delta}\}$ the set of sparse trees, the above equation yields $|S| \geq \frac{k_l\delta}{2^6}$. The same arguments as in Proposition 3.3 give

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |S| \geq \frac{k_l\delta}{2^6} \geq \frac{\epsilon\delta}{2^7} t_{k_l}.$$

  The only difference is that we chose c_ϵ so that $2^{2 \cdot \frac{2^7}{\epsilon\delta} - 1} \leq \frac{1}{4c_\epsilon}$ as needed in the original proof.

- We now turn to the case when the majority of input points on which $(1+\delta)$C1NN made a mistake are not in the ball $B(\bar{x}, r)$, similarly to Step 2. Using the same notion of sparse tree $S := \{k \in G,\ |\mathcal{T}_k| \leq \frac{2^7}{\epsilon\delta}\}$, we have again $|S| \geq \frac{k_l\delta}{2^6}$. We use the same arguments as in the original proof. Suppose $|\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2}$, then we have

$$|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l\delta}{2^7} \geq \frac{\epsilon\delta}{2^8} t_{k_l}.$$

Step 3

In this last step, we suppose again that the majority of input points on which $(1+\delta)$C1NN made a mistake are not in the ball $B(\bar{x}, r)$ but that $|\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) > r\}| < \frac{|S|}{2}$. Therefore, we obtain

$$|\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) = r\}| = |S| - |\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) > r\}| \geq \frac{|S|}{2} \geq \frac{k_l\delta}{2^7} \geq \frac{\epsilon\delta}{2^8} t_{k_l}.$$

We will now make use of the partition $(P_i)_{i\geq 1}$. Because $(n_u)_{u\geq 1}$ is an increasing sequence, let $u \geq 1$ such that $n_{u+1} \leq t_{k_l} \leq n_{u+2}$ (we can suppose without loss of generality that $t_{k_0} > n_2$). Note that we have $n_u \leq \frac{\epsilon\delta}{2^9} n_{u+1} \leq \frac{\epsilon\delta}{2^9} t_{k_l}$. Let us now analyze the process between times n_u and t_{k_l}. In particular, we are interested in the indices $T = \{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) = r\}$ and times $\mathcal{U}_u = \{p^k_{d(k)} :\ n_u < p^k_{d(k)} \leq k_l,\ k \in T\}$. In particular, we have

$$|\mathcal{U}_u| \geq |\{k \in S,\ \rho_{\mathcal{X}}(x_{p^k_{d(k)}}, \bar{x}) = r\}| - n_u \geq \frac{\epsilon\delta}{2^8} t_{k_l} - \frac{\epsilon\delta}{2^9} t_{k_l} = \frac{\epsilon\delta}{2^9} t_{k_l}.$$

Defining $T' := \{k \in T,\ r - \frac{r}{2^{u+3}} \leq \rho_{\mathcal{X}}(x_{\phi(t_k)}, \bar{x}) < r\}$, the same arguments as in the original proof yield

$$|\{i,\ P_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq |T'| \geq |\mathcal{U}_u| - |\{i,\ P_i(\tau_u) \cap \boldsymbol{x}_{\mathcal{U}_u} \neq \emptyset\}| \geq \frac{\epsilon\delta}{2^9} t_{k_l} - \frac{\epsilon\delta}{2^{10}} t_{k_l} = \frac{\epsilon\delta}{2^{10}} t_{k_l}.$$

Step 4

In conclusion, in all cases, we obtain

$$|\{Q \in \mathcal{Q},\ Q \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}| \geq \max(|\{i,\ A_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}|, |\{i,\ P_i \cap \boldsymbol{x}_{\leq t_{k_l}} \neq \emptyset\}|) \geq \frac{\epsilon\delta}{2^{10}} t_{k_l}.$$

Because this is true for all $l \geq \hat{l}$ and t_{k_l} is an increasing sequence, we conclude that $\boldsymbol{x}$ disproves the SMV condition for $\mathcal{Q}$. Recall that this holds whenever the event $\mathcal{A} \cap \mathcal{M} \cap \bigcap_{l\geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)$ is met. Thus,

$$\mathbb{P}[|\{Q \in \mathcal{Q},\ Q \cap \mathbb{X}_{<T}\}| = o(T)] \leq 1 - \mathbb{P}\left[\mathcal{A} \cap \mathcal{M} \cap \bigcap_{l\geq 1}(\mathcal{E}_l \cap \mathcal{F}_l)\right] \leq 1 - \frac{\eta}{4} < 1.$$

This shows that $\mathbb{X} \notin$ SMV which is absurd. Therefore $(1+\delta)$C1NN is consistent on f^*. This ends the proof of the proposition. ■

Using the fact that in the $(1+\delta)$C1NN learning rule, no time t can have more than 2 children, as the 2C1NN rule, we obtain with the same proof as in Chapter 3 (Theorem 3.8) the following proposition.

Proposition 4.3. *Fix $0 < \delta \leq 1$. Let $(\mathcal{X}, \mathcal{B})$ be a separable Borel space. For the binary classification setting, the learning rule $(1+\delta)$C1NN is universally consistent for all processes $\mathbb{X} \in$ SMV.*

Finally, we use the reduction in Proposition 3.5 from Chapter 3 which gives a reduction from any near-metric bounded value space to binary classification.

Proposition 4.4. *If $(1+\delta)$C1NN is universally consistent under a process $\mathbb{X}$ for binary classification, it is also universally consistent under $\mathbb{X}$ for any separable near-metric setting $(\mathcal{Y}, \ell)$ with bounded loss.*

Together with Proposition 4.3, Proposition 4.4 ends the proof of Theorem 4.5.

4.9.2 Proof of Theorem 4.6

Let $0 < \epsilon \leq 1$. We first analyze the prediction of the learning rule $f^{\epsilon}_{\cdot}$. In the rest of the proof, we denote $\bar{\ell}(\hat{Y}_t(\epsilon), Y_t) := \sum_{y \in \mathcal{Y}_\epsilon} \mathbb{P}(\hat{Y}_t(\epsilon) = y)\ell(y, Y_t)$ the immediate expected loss at each iteration. The learning rule was constructed so that we perform exactly the classical Hedge / exponentially weighted average forecaster on each cluster of times $\mathcal{C}(t) = \{u \leq t : u \overset{\phi}{\sim} t\}$. As a result [CL06] (Theorem 2.2), we have that for any $t \geq 1$,

$$\begin{aligned}
\frac{1}{\bar{\ell}} \sum_{u \in \mathcal{C}(t)} \bar{\ell}(\hat{Y}_u(\epsilon), Y_u) &\leq \frac{1}{\bar{\ell}} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}(t)} \ell(y, Y_u) + \frac{\ln |\mathcal{Y}_\epsilon|}{\bar{\ell}\eta_\epsilon} + \frac{|\mathcal{C}(t)|\bar{\ell}\eta_\epsilon}{8} \\
&\leq \frac{1}{\bar{\ell}} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}(t)} \ell(y, Y_u) + \sqrt{\frac{\ln |\mathcal{Y}_\epsilon|}{8T_\epsilon}}(T_\epsilon + |\mathcal{C}(t)|) \\
&\leq \frac{1}{\bar{\ell}} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}(t)} \ell(y, Y_u) + \frac{\epsilon}{\bar{\ell}} \max(T_\epsilon, |\mathcal{C}(t)|)
\end{aligned}$$

Now consider a horizon $T \geq 1$, and enumerate all the clusters $\mathcal{C}_1(T), \ldots, \mathcal{C}_{p(T)}(T)$ at horizon T, i.e. the classes of equivalence of ϕ among the times $\{t \leq T\}$. Note that if a cluster $i \leq p$ has $|\mathcal{C}_i(T)| < T_\epsilon$, then either it must contain a time $t \in \mathcal{N}$ which is a leaf of the tree formed by ϕ until time T, or it is a cluster of duplicates of an instance X_u which has already had $\frac{T_\epsilon}{\epsilon}$ occurrences. As a result, the times falling into such clusters of duplicates with less than T_ϵ members form at most a proportion ϵ of the total T times. Denote by $\mathcal{A}_i := \{t \leq T : t \in \mathcal{N}, |\{u \leq T : \phi(u) = t\}| = i\}$ times which have excactly i children for $i \in \{0, 1, 2\}$. Note that no time can have more than 2 children. In particular $\mathcal{A}_0$ is the set of leaves. Then, by summing the above equations we obtain

$$\begin{aligned}
\sum_{t=1}^{T} \bar{\ell}(\hat{Y}_t(\epsilon), Y_t) &\leq \sum_{i=1}^{p(T)} \left(\min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + \epsilon \max(T_\epsilon, |\mathcal{C}_i(T)|) \right) \\
&\leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + \epsilon T + T_\epsilon |\{1 \leq i \leq p : |\mathcal{C}_i(T)| < T_\epsilon\}| \\
&\leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + \epsilon T + T_\epsilon |\mathcal{A}_0| + \epsilon T_\epsilon,
\end{aligned}$$

where in the last inequality we used the fact that all clusters with $|\mathcal{C}_i(T)| < T_\epsilon$ contain a leaf from $\mathcal{A}_0$, which is therefore distinct for each such cluster. Now note that by counting the number of edges of the tree structure we obtain $\frac{1}{2}(3|\mathcal{A}_2| + 2|\mathcal{A}_1| + |\mathcal{A}_0| - 1) = T - 1 = |\mathcal{A}_0| + |\mathcal{A}_1| + |\mathcal{A}_2| - 1$, where the -1 on the left-hand side accounts for the root of this tree which does not have a parent. Hence we obtain $|\mathcal{A}_0| = |\mathcal{A}_2| + 1$. Further, $|\mathcal{A}_2| \leq |\{t \leq T : U_t = 1\}|$ which follows a binomial distribution $\mathcal{B}(T, \delta_\epsilon)$. Therefore, using the Chernoff bound, with

probability $1 - e^{-T\delta_\epsilon/3}$ we have

$$\sum_{t=1}^{T} \bar{\ell}(\hat{Y}_t(\epsilon), Y_t) \leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + 2\epsilon T + T_\epsilon(1 + 2T\delta_\epsilon)$$
$$\leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + T_\epsilon + 3\epsilon T.$$

We now observe that the sequence $\{\ell(\hat{Y}_t(\epsilon), Y_t) - \bar{\ell}(\hat{Y}_t(\epsilon), Y_t)\}_{T \geq 1}$ is a sequence of martingale differences bounded by $\bar{\ell}$ in absolute value. Hence, the Hoeffding-Azuma inequality yields that for any $T \geq 1$, with probability $1 - \frac{1}{T^2} - e^{-T\delta_\epsilon/3}$,

$$\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon), Y_t) \leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + T_\epsilon + 3\epsilon T + 2\bar{\ell}\sqrt{T \ln T}.$$

Because $\sum_{T \geq 1} \frac{1}{T^2} + e^{-T\delta_\epsilon/3} < \infty$ the Borel-Cantelli lemma implies that with probability one, there exists a time $\hat{T}$ such that

$$\forall T \geq \hat{T}, \quad \sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon), Y_t) \leq \sum_{i=1}^{p(T)} \min_{y \in \mathcal{Y}_\epsilon} \sum_{u \in \mathcal{C}_i(T)} \ell(y, Y_u) + T_\epsilon + 2\bar{\ell}\sqrt{T \ln T} + 3\epsilon T.$$

We denote by $\mathcal{E}_\epsilon$ this event. We are now ready to analyze the risk of the learning rule $f^\epsilon_\cdot$. Let $f : \mathcal{X} \to \mathcal{Y}$ a measurable function to which we compare the prediction of $f^\epsilon_\cdot$. By Theorem 4.5, the rule $(1+\delta_\epsilon)$C1NN is optimistically universal in the noiseless setting. Therefore, because $\mathbb{X} \in$ Soul we have in particular

$$\frac{1}{T} \sum_{t=1}^{T} \ell((1+\delta_\epsilon)C1NN_t(\mathbb{X}_{\leq t-1}, f(\mathbb{X}_{\leq t-1}), X_t), f(X_t)) \to 0 \quad (a.s.),$$

i.e., almost surely, $\frac{1}{T}\sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi(t)}), f(X_t)) \to 0$ — the times corresponding to duplicate instances incur a 0 loss by memorization. We denote by $\mathcal{F}_\epsilon$ this event of probability one. We write for any $u = 1, \ldots, T_\epsilon - 1$,

$$\sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi^u(t)}), f(X_t))$$
$$\leq c^\ell_1 \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi^{u-1}(t)}), f(X_t)) + 2 \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi^l(t)}), f(X_{\phi^{u-1}(t)}))$$
$$\leq c^\ell_1 \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi^{u-1}(t)}), f(X_t))$$
$$+ 2 \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi(t)}), f(X_t)) \cdot |\{l \leq T : \phi^{u-1}(l) = t\}|$$
$$\leq c^\ell_1 \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi^{u-1}(t)}), f(X_t)) + 2^u \sum_{t \leq T, t \in \mathcal{N}} \ell(f(X_{\phi(t)}), f(X_t))$$

where we used the fact that times have at most 2 children. Therefore, iterating the above equations, we obtain that on $\mathcal{F}_\epsilon$, for any $u = 1, \dots, T_\epsilon - 1$

$$\begin{aligned}\frac{1}{T}\sum_{t\leq T, t\in\mathcal{N}} \ell(f(X_{\phi^u(t)}), f(X_t)) &\leq \left(\sum_{k=1}^{u}(c_1^\ell)^{u-k}2^k\right)\frac{1}{T}\sum_{t\leq T, t\in\mathcal{N}} \ell(f(X_{\phi(t)}), f(X_t)) \\ &\leq \frac{u2^u(c_1^\ell)^u}{T}\sum_{t\leq T, t\in\mathcal{N}} \ell(f(X_{\phi(t)}), f(X_t)) \to 0.\end{aligned}$$

In the rest of the proof, for any $y \in \mathcal{Y}$, we will denote by y^ϵ a value in the $\epsilon-$net $\mathcal{Y}_\epsilon$ such that $\ell(y, y^\epsilon) \leq \epsilon$. We now pose $\mu_\epsilon = \min\{0 < \mu \leq 1 : c_\mu^\ell \leq \frac{1}{\sqrt{\epsilon}}\}$ if the corresponding set is non-empty and $\mu_\epsilon = 1$ otherwise. Note that because c_μ^ℓ is non-increasing in μ, we have $\mu_\epsilon \longrightarrow_{\epsilon\to 0} 0$. Finally, for any cluster $\mathcal{C}_i(T)$, let $t_i = \min\{u \in \mathcal{C}_i(T)\}$. Putting everything together, on the event $\mathcal{E}_\epsilon \cap \mathcal{F}_\epsilon$, for any $T \geq \hat{T}$, we have

$$\begin{aligned}\sum_{t=1}^{T}\ell(\hat{Y}_t(\epsilon), Y_t) &\leq \sum_{i=1}^{p(T)}\min_{y\in\mathcal{Y}_\epsilon}\sum_{u\in\mathcal{C}_i(T)}\ell(y, Y_u) + T_\epsilon + 2\bar{\ell}\sqrt{T\ln T} + 3\epsilon T \\ &\leq \sum_{i=1}^{p(T)}\sum_{u\in\mathcal{C}_i(T)}\ell(f(X_{t_i})^\epsilon, Y_u) + T_\epsilon\bar{\ell} + 2\bar{\ell}\sqrt{T\ln T} + 3\epsilon T \\ &\leq \sum_{i=1}^{p(T)}\sum_{u\in\mathcal{C}_i(T)}[c_{\mu_\epsilon}^\ell\ell(f(X_{t_i})^\epsilon, f(X_{t_i})) + (c_{\mu_\epsilon}^\ell)^2\ell(f(X_{t_i}), f(X_u)) \\ &\qquad + (1+\mu_\epsilon)^2\ell(f(X_u), Y_u)] + T_\epsilon\bar{\ell} + 2\bar{\ell}\sqrt{T\ln T} + 3\epsilon T \\ &\leq (1+\mu_\epsilon)^2\sum_{t=1}^{T}\ell(f(X_t), Y_t) + (c_{\mu_\epsilon}^\ell)^2\frac{T_\epsilon}{\epsilon}\sum_{u=1}^{T_\epsilon-1}\sum_{t\leq T, t\in\mathcal{N}}\ell(f(X_t), f(X_{\phi^u(t)})) \\ &\qquad + T_\epsilon\bar{\ell} + 2\bar{\ell}\sqrt{T\ln T} + (3 + c_{\mu_\epsilon}^\ell)\epsilon T \\ &\leq \sum_{t=1}^{T}\ell(f(X_t), Y_t) + \frac{(c_{\mu_\epsilon}^\ell)^2 T_\epsilon}{\epsilon}\sum_{u=1}^{T_\epsilon-1}\sum_{t\leq T, t\in\mathcal{N}}\ell(f(X_t), f(X_{\phi^u(t)})) \\ &\qquad + T_\epsilon\bar{\ell} + 2\bar{\ell}\sqrt{T\ln T} + (3\epsilon + \epsilon c_{\mu_\epsilon}^\ell + 3\mu_\epsilon)T,\end{aligned}$$

where in the third inequality we used the generalized-metric loss identity twice, and in the fourth inequality we used the fact that clusters containing distinct instances have at most $\frac{T_\epsilon}{\epsilon}$ duplicates of each instance. Hence, for any $\epsilon < (c_1^\ell)^{-2}$, on the event $\mathcal{E}_\epsilon \cap \mathcal{F}_\epsilon$, we obtain

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t(\epsilon), Y_t) - \ell(f(X_t), Y_t) \leq 3\epsilon + \epsilon c_{\mu_\epsilon}^\ell + 3\mu_\epsilon \leq 3\epsilon + \sqrt{\epsilon} + 3\mu_\epsilon,$$

where $\mu_\epsilon \longrightarrow_{\epsilon\to 0} 0$. We now denote $\delta_\epsilon := 2\epsilon + \sqrt{\epsilon} + 3\mu_\epsilon$ and $i_0 = \lceil\frac{2\ln c_1^\ell}{\ln 2}\rceil$. We now turn to the final learning rule and show that by using the predictions of the rules $f^{\epsilon_i}_\cdot$ for $i \geq 0$, it

achieves zero risk. First, by the union bound, on the event $\bigcap_{i\geq 0} \mathcal{E}_{\epsilon_i} \cap \mathcal{F}_{\epsilon_i}$ of probability one,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) \leq \delta_{\epsilon_i}, \quad \forall i \geq i_0.$$

Now define $\mathcal{H}$ the event probability one according to Lemma 4.1 such that there exists $\hat{t}$ for which

$$\forall t \geq \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^{t} \ell(\hat{Y}_t, Y_t) \leq \sum_{s=t_i}^{t} \ell(\hat{Y}_t(\epsilon_i), Y_t) + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{t \ln t}.$$

In the rest of the proof we will suppose that the event $\mathcal{H} \cap \bigcap_{i\geq 0} \mathcal{E}_{\epsilon_i} \cap \mathcal{F}_{\epsilon_i}$ is met. Let $i \geq i_0$. For any $T \geq \max(\hat{t}, t_i)$, we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) &\leq \frac{t_i}{T}\bar{\ell} + \frac{1}{T}\sum_{t=t_i}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \\
&\leq \frac{t_i}{T}\bar{\ell} + \frac{1}{T}\sum_{t=t_i}^{T} \ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{\frac{\ln T}{T}} \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) + \frac{2t_i}{T}\bar{\ell} + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{\frac{\ln T}{T}}.
\end{aligned}$$

Therefore we obtain $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq \delta_{\epsilon_i}$. Because this holds for any $i \geq i_0$ on the event $\mathcal{H} \cap \bigcap_{i\geq 0} \mathcal{E}_{\epsilon_i} \cap \mathcal{F}_{\epsilon_i}$ of probability one, and $\delta_{\epsilon_i} \to 0$ for $i \to \infty$, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 0.$$

This ends the proof of the theorem.

4.9.3 Proof of Lemma 4.1

We first introduce the following helper lemma which can be found in [CL06].

Lemma 4.6 ([CL06]). *For all $N \geq 2$, for all $\beta \geq \alpha \geq 0$ and for all $d_1, \ldots, d_N \geq 0$ such that $\sum_{i=1}^{N} e^{-\alpha d_i} \geq 1$,*

$$\ln \frac{\sum_{i=1}^{N} e^{-\alpha d_i}}{\sum_{i=1}^{N} e^{-\beta d_i}} \leq \frac{\beta - \alpha}{\alpha} \ln N.$$

We are now ready to compare the predictions of the learning rule $f_\cdot$ to the predictions of the rules $f^\epsilon_\cdot$.

For any $t \geq 0$, we define the instantaneous regret $r_{t,i} = \hat{\ell}_t - \ell(\hat{Y}_t(\epsilon_i), Y_t)$. We first note that $|r_{t,i}| \leq \bar{\ell}$. We now define $w'_{t-1,i} := e^{\eta_{t-1}(\hat{L}_{t-1,i} - L_{t-1,i})}$. We also introduce $W_{t-1} = \sum_{i\in I_t} w_{t-1,i}$

and $W'_{t-1} = \sum_{i \in I_{t-1}} w'_{t-1,i}$. We denote the index $k_t \in I_t$ such that $\hat{L}_{t,k_t} - L_{t,k_t} = \max_{i \in I_t} \hat{L}_{t,i} - L_{t,i}$. Then we write

$$\frac{1}{\eta_t} \ln \frac{w_{t-1,k_{t-1}}}{W_{t-1}} - \frac{1}{\eta_{t+1}} \ln \frac{w_{t,k_t}}{W_t} = \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) \ln \frac{W_t}{w_{t,k_t}} + \frac{1}{\eta_t} \ln \frac{W_t / w_{t,k_t}}{W'_t / w'_{t,k_t}} + \frac{1}{\eta_t} \ln \frac{w_{t-1,k_{t-1}}}{w'_{t,k_t}} + \frac{1}{\eta_t} \ln \frac{W'_t}{W_{t-1}}.$$

By construction, we have $\ln \frac{W_t}{w_{t,k_t}} \leq \ln |I_t| \leq \ln(1 + \ln t)$. Further, we have that

$$\begin{aligned}
\frac{1}{\eta_t} \ln \frac{W_t / w_{t,k_t}}{W'_t / w'_{t,k_t}} &= \frac{1}{\eta_t} \ln \frac{\sum_{i \in I_{t+1}} e^{\eta_{t+1}(\hat{L}_{t,i} - L_{t,i} - \hat{L}_{t,k_t} + L_{t,k_t})}}{\sum_{i \in I_t} e^{\eta_t(\hat{L}_{t,i} - L_{t,i} - \hat{L}_{t,k_t} + L_{t,k_t})}} \\
&= \frac{1}{\eta_t} \ln \frac{\sum_{i \in I_{t+1}} w_{t,i}}{\sum_{i \in I_t} w_{t,i}} + \frac{1}{\eta_t} \ln \frac{\sum_{i \in I_{t+1}} e^{\eta_{t+1}(\hat{L}_{t,i} - L_{t,i} - \hat{L}_{t,k_t} + L_{t,k_t})}}{\sum_{i \in I_{t+1}} e^{\eta_t(\hat{L}_{t,i} - L_{t,i} - \hat{L}_{t,k_t} + L_{t,k_t})}} \\
&\leq \frac{1}{\eta_t} \ln \frac{\sum_{i \in I_{t+1}} w_{t,i}}{\sum_{i \in I_t} w_{t,i}} + \frac{1}{\eta_t} \left(\frac{\eta_t - \eta_{t+1}}{\eta_{t+1}} \right) \ln |I_{t+1}| \\
&\leq \frac{|I_{t+1}| - |I_t|}{\eta_t \sum_{i \in I_t} w_{t,i}} + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) \ln(1 + \ln(t+1)),
\end{aligned}$$

where in the first inequality we applied Lemma 4.6. We also have

$$\frac{1}{\eta_t} \ln \frac{w_{t-1,k_{t-1}}}{w'_{t,k_t}} = (\hat{L}_{t-1,k_{t-1}} - L_{t-1,k_{t-1}}) - (\hat{L}_{t,k_t}, L_{t,k_t}).$$

Last, because $|r_{t,i}| \leq \bar{\ell}$ for all $i \in I_t$, we can use Hoeffding's lemma to obtain

$$\frac{1}{\eta_t} \ln \frac{W'_t}{W_{t-1}} = \frac{1}{\eta_t} \ln \sum_{i \in I_t} \frac{w_{t-1,i}}{W_{t-1}} e^{\eta_t r_{t,i}} \leq \frac{1}{\eta_t} \left(\eta_t \sum_{i \in I_t} r_{t,i} \frac{w_{t-1,i}}{W_{t-1}} + \frac{\eta_t^2 (2\bar{\ell})^2}{8} \right) = \frac{1}{2} \eta_t \bar{\ell}^2.$$

Putting everything together gives

$$\frac{1}{\eta_t} \ln \frac{w_{t-1,k_{t-1}}}{W_{t-1}} - \frac{1}{\eta_{t+1}} \ln \frac{w_{t,k_t}}{W_t} \leq 2 \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) \ln(1 + \ln(t+1)) + \frac{|I_{t+1}| - |I_t|}{\eta_t \sum_{i \in I_t} w_{t,i}} + (\hat{L}_{t-1,k_{t-1}} - L_{t-1,k_{t-1}}) - (\hat{L}_{t,k_t} - L_{t,k_t}) + \frac{1}{2} \eta_t \bar{\ell}^2. \quad (4.1)$$

First suppose that we have $\sum_{i \in I_t} w_{t,i} \leq 1$. Then either $k_t \in I_{t+1} \setminus I_t$ in which case $\hat{L}_{t,k_t} - L_{t,k_t} = 0$, or we have directly

$$\hat{L}_{t,k_t} - L_{t,k_t} \leq \frac{1}{\eta_{t+1}} \ln \left[\sum_{i \in I_t} w_{t,i} \right] \leq 0.$$

Otherwise, let $t' = \min\{1 \le s \le t : \forall s \le s' \le t, \sum_{i \in I_{s'}} w_{s',i} \ge 1\}$. We sum Eq (4.1) for $s = t', \dots, t$ which gives

$$\frac{1}{\eta_1} \ln \frac{w_{t'-1,k_{t'-1}}}{W_{t'-1}} - \frac{1}{\eta_{t+1}} \ln \frac{w_{t,k_t}}{W_t} \le \frac{2}{\eta_{t+1}} \ln(1 + \ln(t+1)) + \frac{|I_{t+1}|}{\eta_t} + (\hat{L}_{t'-1,k_{t'-1}} - L_{t'-1,k_{t'-1}}) - (\hat{L}_{t,k_t} - L_{t,k_t}) + \frac{\bar{\ell}^2}{2} \sum_{s=t'}^{t} \eta_s.$$

Note that we have $\frac{w_{t,k_t}}{W_t} \le 1$ and $\frac{w_{t'-1,k_{t'-1}}}{W_{t'-1}} \ge \frac{1}{|I_{t'-1}|} \ge \frac{1}{1+\ln t}$. Also, assuming $t' \ge 2$, since $\sum_{i \in I_{t'-1}} w_{t'-1,i} < 1$, we have for any $i \in I_{t'-1}$ that $\hat{L}_{t'-1,i} - L_{t'-1,i} \le 0$, hence $\hat{L}_{t'-1,k_{t'-1}} - L_{t'-1,k_{t'-1}} \le 0$. If $t' = 1$ we have directly $\hat{L}_{0,k_0} - L_{0,k_0} = 0$. Finally, using the fact that $\sum_{s=1}^{t} \frac{1}{\sqrt{s}} \le 2\sqrt{t}$, we obtain

$$\begin{aligned}\hat{L}_{t,k_t} - L_{t,k_t} &\le \ln(1 + \ln(t+1)) \left(1 + 2\sqrt{\frac{t+1}{\ln(t+1)}}\right) + (1 + \ln(t+1))\sqrt{\frac{t}{\ln t}} + \bar{\ell}^2 \sqrt{t \ln t} \\ &\le (3/2 + \bar{\ell}^2)\sqrt{t \ln t},\end{aligned}$$

for all $t \ge t_0$ where t_0 is a fixed constant. This in turn implies that for all $t \ge t_0$ and $i \in I_t$, we have $\hat{L}_{t,i} - L_{t,i} \le (3/2 + \bar{\ell}^2)\sqrt{t \ln t}$. Now note that $|\ell(\hat{Y}_t, Y_t) - \hat{\ell}_t| \le \bar{\ell}$. Hence, we can use Hoeffding-Azuma inequality for the variables $\ell(\hat{Y}_t, Y_t) - \hat{\ell}_t$ that form a sequence of martingale differences to obtain $\mathbb{P}\left[\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) > \hat{L}_{t,i} + u\right] \le e^{-\frac{2u^2}{t\bar{\ell}^2}}$. Hence, for $t \ge t_0$ and $i \in I_t$, with probability $1 - \delta$, we have

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \le \hat{L}_{t,i} + \bar{\ell}\sqrt{\frac{t}{2} \ln \frac{1}{\delta}} \le L_{t,i} + (3/2 + \bar{\ell}^2)\sqrt{t \ln t} + \bar{\ell}\sqrt{\frac{t}{2} \ln \frac{1}{\delta}}.$$

Therefore, since $|I_t| \le 1 + \ln t$, by union bound with probability $1 - \frac{1}{t^2}$ we obtain that for all $i \in I_t$,

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \le L_{t,i} + (3/2 + \bar{\ell}^2)\sqrt{t \ln t} + \bar{\ell}\sqrt{\frac{t}{2} \ln(1 + \ln t)} + \bar{\ell}\sqrt{t \ln t} \le (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{t \ln t},$$

for all $t \ge t_1$ where $t_1 \ge t_0$ is a fixed constant. The Borel-Cantelli lemma implies that almost surely, there exists $\hat{t} \ge 0$ such that

$$\forall t \ge \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \le L_{t,i} + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{t \ln t}.$$

This ends the proof of the lemma.

4.10 Appendix B: Proofs of Section 4.4

4.10.1 Proof of Theorem 4.7

We start by checking that with the defined loss $(\mathbb{N}, \ell)$ is indeed a metric space $(\mathbb{N}, \ell)$. We only have to check that the triangular inequality is satisfied, the other properties of a metric being directly satisfied. By construction, the loss has values in $\{0, \frac{1}{2}, 1\}$. Now let $i, j, k \in \mathbb{N}$. The triangular inequality $\ell(i,j) \leq \ell(i,k) + \ell(k,j)$ is directly satisfied if two of these indices are equal. Therefore, we can suppose that they are all distinct and as a result $\ell(i,j), \ell(i,k), \ell(k,j) \in \{\frac{1}{2}, 1\}$. Therefore

$$\ell(i,j) \leq 1 \leq \ell(i,k) + \ell(k,j),$$

which ends the proof that ℓ is a metric.

Now let $(\mathcal{X}, \mathcal{B})$ be a separable metrizable Borel space. Let $\mathbb{X} \notin \mathrm{CS}$. We aim to show that universal online learning under adversarial responses is not achievable under $\mathbb{X}$. Because $\mathbb{X} \notin \mathrm{CS}$, there exists a sequence of decreasing measurable sets $\{A_i\}_{i\geq 1}$ with $A_i \downarrow \emptyset$ such that $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_i)]$ does not converge to 0 for $i \to \infty$. In particular, there exist $\epsilon > 0$ and an increasing subsequence $(i_l)_{l\geq 1}$ such that $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_{i_l})] \geq \epsilon$ for all $l \geq 1$. We now denote $B_l := A_{i_l} \setminus A_{i_{l+1}}$ for any $l \geq 1$. Then $\{B_l\}_{l\geq 1}$ forms a sequence of disjoint measurable sets such that

$$\mathbb{E}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right)\right] \geq \epsilon, \quad l \geq 1.$$

Therefore, for any $l \geq 1$ because $\mathbb{E}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right)\right] \leq \mathbb{P}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right) \geq \frac{\epsilon}{2}\right] + \frac{\epsilon}{2}$ we obtain

$$\mathbb{P}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right) \geq \frac{\epsilon}{2}\right] \geq \frac{\epsilon}{2}.$$

Now because $\hat{\mu}$ is increasing we obtain

$$\begin{aligned}\mathbb{P}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right) \geq \frac{\epsilon}{2}, \forall l \geq 1\right] &= \lim_{L\to\infty} \mathbb{P}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right) \geq \frac{\epsilon}{2}, 1 \leq l \leq L\right] \\ &= \lim_{L\to\infty} \mathbb{P}\left[\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq L} B_{l'}\right) \geq \frac{\epsilon}{2}\right] \geq \frac{\epsilon}{2}.\end{aligned}$$

We will denote by $\mathcal{A}$ this event in which for all $l \geq 1$, we have $\hat{\mu}_{\mathbb{X}}\left(\bigcup_{l'\geq l} B_{l'}\right) \geq \frac{\epsilon}{2}$. Under the event $\mathcal{A}$, for any $l, t^0 \geq 1$, there always exists $t^1 > t^0$ such that $\frac{1}{t^1}\sum_{t=1}^{t^1} \mathbb{1}_{\bigcup_{l'\geq l} B_{l'}}(X_t) \geq \frac{3\epsilon}{8}$. We construct a sequence of times $(t_p)_{p\geq 1}$ and indices $(l_p)_{p\geq 1}$, $(u_p)_{p\geq 1}$ by induction as follows. We first pose $u_0 = t_0 = 0$. Now assume that for $p \geq 1$, the time t_{p-1} and index u_{p-1} are defined. We first construct an index $l_p > u_{p-1}$ such that

$$\mathbb{P}\left[\mathbb{X}_{\leq t_{p-1}} \cap \left(\bigcup_{l\geq l_p} B_l\right) \neq \emptyset\right] \leq \frac{\epsilon}{2^{p+3}}.$$

We will denote by $\mathcal{E}_p$ the complementary of this event. Note that finding such index l_p is possible because the considered events $\{\mathbb{X}_{\leq t_{p-1}} \cap \left(\bigcup_{l' \geq l} B_{l'}\right) \neq \emptyset\}$ are decreasing as $l > u_{p-1}$ increases and we have

$$\bigcap_{l > u_{p-1}} \left\{ \mathbb{X}_{\leq t_{p-1}} \cap \left(\bigcup_{l' \geq l} B_{l'} \right) \neq \emptyset \right\} = \left\{ \mathbb{X}_{\leq t_{p-1}} \cap \left(\bigcap_{l > u_{p-1}} \bigcup_{l' \geq l} B_{l'} \right) \neq \emptyset \right\} = \emptyset.$$

We then construct $t_p > t_{p-1}$ such that

$$\mathbb{P}\left[\mathcal{A}^c \cup \bigcup_{t_{p-1} < t \leq t_p} \left\{ \frac{1}{t} \sum_{u=1}^{t} \mathbb{1}_{\bigcup_{l \geq l_p} B_l}(X_u) \geq \frac{3\epsilon}{8} \right\} \right] \geq 1 - \frac{\epsilon}{2^{p+4}}.$$

This is also possible because $\mathcal{A} \subset \bigcup_{t > \frac{8}{\epsilon} t_{p-1}} \left\{ \frac{1}{t} \sum_{u=1}^{t} \mathbb{1}_{\bigcup_{l \geq l_p} B_l}(X_u) \geq \frac{3\epsilon}{8} \right\}$. Last, we can now construct $u_p \geq l_p$ such that

$$\mathbb{P}\left[\mathcal{A}^c \cup \bigcup_{t_{p-1} < t \leq t_p} \left\{ \frac{1}{t} \sum_{u=1}^{t} \mathbb{1}_{\bigcup_{l_p \leq l \leq u_p} B_l}(X_u) \geq \frac{\epsilon}{4} \right\} \right] \geq 1 - \frac{\epsilon}{2^{p+3}},$$

which is possible using similar arguments as above. We denote $\mathcal{F}_p$ this event. This ends the recursive construction of times t_p and indices l_p for all $p \geq 1$. Note that by construction, $\mathbb{P}[\mathcal{E}_p^c], \mathbb{P}[\mathcal{F}_p^c] \leq \frac{\epsilon}{2^{p+3}}$. Hence, by union bound, the event $\mathcal{A} \cap \bigcap_{p \geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)$ has probability $\mathbb{P}[\mathcal{A} \cap \bigcap_{p \geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)] \geq \mathbb{P}[\mathcal{A}] - \frac{\epsilon}{4} \geq \frac{\epsilon}{4}$. To simplify the rest of the proof, we denote $\tilde{B}_p = \bigcup_{l_p \leq l \leq u_p} B_l$ for any $p \geq 1$. Also, for any $t_1 \leq t_2$, we denote by

$$N_p(t_1, t_2) = \sum_{t=t_1}^{t_2} \mathbb{1}_{\tilde{B}_p}(X_t)$$

the number of times that set $\tilde{B}_p$ has been visited between times t_1 and t_2.

We now fix a learning rule f. and construct a process $\mathbb{Y}$ for which consistency will not be achieved on the event $\mathcal{A} \cap \bigcap_{p \geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)$. Precisely, we first construct a family of processes $\mathbb{Y}^b$ indexed by a sequence of binary digits $b = (b_i)_{i \geq 1}$. The process $\mathbb{Y}^b$ is defined such that for any $p \geq 1$, and for all $t_{p-1} < t \leq t_p$,

$$Y_t^b := \begin{cases} n_{t_p} + 4u_p(t) + 2b_{i(p,u_p(t))} + b_{i(p,u_p(t))+1} & \text{if } X_t \in \tilde{B}_p, \\ n_{t_{p'}} + 4t_{p'} + \{b_{i(p',t_{p'}-1)} \dots b_{i(p',1)} b_{i(p',0)}\}_2 & \text{if } X_t \in \tilde{B}_{p'}, p' < p, \\ 0 & \text{otherwise,} \end{cases}$$

where we denoted $u_p(t) = N_p(t_{p-1} + 1, t - 1)$ and posed for any $p \geq 1$ and $u \geq 1$:

$$i(p, u) = 2 \sum_{p' < p} t_{p'} + 2u.$$

Note in particular that conditionally on $\mathbb{X}$, $\mathbb{Y}^b$ is deterministic: it does not depends on the random predictions of the learning rule. Because we always have $N_p(t_{p-1} + 1, t - 1) \leq t_p$

for any $t \le t_p$, the process is designed so that we have $Y_t^b \in I_{t_p}$ if $X_t \in \tilde{B}_p$ and $t_{p-1} < t \le t_p$. Further, for $t_{p-1} < t \le t_p$, if $X_t \in \bigcup_{p'<p} \tilde{B}_{p'}$ then $Y_t^b \in J_{t_{p'}}$. We now consider an i.i.d. Bernoulli $\mathcal{B}(\frac{1}{2})$ sequence of random bits $\boldsymbol{b}$ independent from the process $\mathbb{X}$—and any learning rule predictions. We analyze the responses of the learning rule for responses $\mathbb{Y}^{\boldsymbol{b}}$. We first fix a realization $\boldsymbol{x}$ of the process $\mathbb{X}$, which falls in the event $\mathcal{A} \cap \bigcap_{p\ge1}(\mathcal{E}_p \cap \mathcal{F}_p)$. For any $p \ge 1$ we define $\mathcal{T}_p := \{t_{p-1} < t \le t_p : \ x_t \in \tilde{B}_p\}$. For simplicity of notation, for any $t \in \mathcal{T}_p$ we denote $i(t) = i(p, u_p(t))$. We will also denote $\hat{Y}_t := f_t(\boldsymbol{x}_{<t}, \mathbb{Y}^{\boldsymbol{b}}_{<t}, x_t)$. Last, denote by r_t the possible randomness used by the learning rule f_t at time t. For any $t \in \mathcal{T}_p$, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{b},\boldsymbol{r}}\ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) &= \mathbb{E}_{\{b_{i(p',u')}, b_{i(p',u')+1},\, p'\le p, u'\le t_{p'}\}\cup\{r_{t'}, t'\le t\}}\ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) \\
&= \mathbb{E}\left[\mathbb{E}_{b_{i(t)}, b_{i(t)+1}}\, \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) \middle|\, b_{i(t')}, b_{i(t')+1}, t' < t, t' \in \mathcal{T}_p;\ b_i, i < i(p,0);\ r_{t'}, t' \le t\right] \\
&= \mathbb{E}\left[\mathbb{E}_{b_{i(t)}, b_{i(t)+1}}\, \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) \middle|\, \hat{Y}_t\right] \\
&= \mathbb{E}_{\hat{Y}_t}\left[\frac{1}{4}\sum_{m=0}^{3} \ell(\hat{Y}_t, n_{t_p} + 4u_p(t) + m)\right] \\
&= \mathbb{E}_{\hat{Y}_t}\left[\mathbb{1}_{\hat{Y}_t \notin \{n_{t_p}+4u_p(t)+m, 0\le m\le 3\}\cup J_{t_p}} + \frac{3}{4}\mathbb{1}_{\hat{Y}_t \in \{n_{t_p}+4u_p(t)+m, 0\le m\le 3\}} + \frac{3}{4}\mathbb{1}_{\hat{Y}_t \in J_{t_p}}\right] \\
&\ge \frac{3}{4}.
\end{aligned}$$

where in the last equality, we used the fact that if $j \in J_{k(t)}$ then by construction $\ell(j, n_{t_p} + 4u_p(t)) = \ell(j, n_{t_p} + 4u_p(t) + 1)$, $\ell(j, n_{t_p} + 4u_p(t) + 2) = \ell(j, n_{t_p} + 4u_p(t) + 3)$, and $\{\ell(j, n_{t_p} + 4u_p(t)), \ell(j, n_{t_p} + 4u_p(t) + 2)\} = \{\frac{1}{2}, 1\}$. Summing all equations, we obtain for any $t_{p-1} < T \le t_p$,

$$\mathbb{E}_{\boldsymbol{b},\boldsymbol{r}}\left[\sum_{t=1}^{T} \ell(f_t(\boldsymbol{x}_{<t}, \mathbb{Y}^{\boldsymbol{b}}_{<t}, x_t), Y_t^{\boldsymbol{b}})\right] \ge \frac{3}{4}\sum_{p'<p} |\mathcal{T}_{p'}| + \frac{3}{4}|\mathcal{T}_p \cap \{t \le T\}|.$$

This holds for all $p \ge 1$. Let us now compare this loss to the best prediction of a fixed measurable function. Specifically, for any binary sequence b, we consider the following function $f^b : \mathcal{X} \to \mathbb{N}$:

$$f^b(x) = \begin{cases} n_{t_p} + 4t_p + \{b_{i(p,t_p-1)} \dots b_{i(p,1)} b_{i(p,0)}\}_2 & \text{if } x \in \tilde{B}_p \\ 0 & \text{if } x \notin \bigcup_{p\ge1} \tilde{B}_p. \end{cases}$$

Now let $t_{p-1} < t \le t_p$ and $p \ge 1$. If $x_t \in \bigcup_{p'<p} \tilde{B}_{p'}$ we have $f^{\boldsymbol{b}}(x_t) = Y_t^{\boldsymbol{b}}$, hence $\ell(f^{\boldsymbol{b}}(x_t), Y_t^{\boldsymbol{b}}) = 0$. Now if $X_t \in \tilde{B}_p$ by construction we have $\ell(f^{\boldsymbol{b}}(x_t), Y_t^{\boldsymbol{b}}) = \frac{1}{2}$. Finally, observe that because the event $\mathcal{E}_{p+1}$ is satisfied by $\boldsymbol{x}$ there does not exist $t_{p-1} < t \le t_p$ such that $t \in \bigcup_{p'>p} \tilde{B}_{p'} \subset \bigcup_{l\ge l_{p+1}} B_l$. As a result, we have $\ell(f^{\boldsymbol{b}}(x_t), Y_t^{\boldsymbol{b}}) = \frac{1}{2}\mathbb{1}_{t\in\mathcal{T}_p}$ for any $t_{p-1} < t \le t_p$. Thus, we obtain for any $t_{p-1} < T \le t_p$,

$$\mathbb{E}_{\boldsymbol{b},\boldsymbol{r}}\left[\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) - \ell(f^{\boldsymbol{b}}(X_t), Y_t^{\boldsymbol{b}})\right] \ge \frac{1}{4}\sum_{p'\le p} |\mathcal{T}_{p'}| + \frac{1}{4}|\mathcal{T}_p \cap \{t \le T\}| \ge \frac{1}{4}|\mathcal{T}_p \cap \{t \le T\}|.$$

Recall that the event $\mathcal{F}_p$ is satisfied by $\boldsymbol{x}$ for any $p \geq 1$. Therefore, there exists a time $t_{p-1} < T_p \leq t_p$ such that $\sum_{t=1}^{T_p} \mathbb{1}_{\tilde{B}_p}(x_t) \geq \frac{\epsilon T_p}{4}$. Then, note that because the event $\mathcal{E}_p$ is satisfied, we have $\sum_{t=1}^{t_{p-1}} \mathbb{1}_{\tilde{B}_p}(x_t) = 0$. Therefore, we obtain $|\mathcal{T}_p \cap \{t \leq T_p\}| \geq \frac{\epsilon T_p}{4}$, and as a result,

$$\mathbb{E}_{\boldsymbol{b},\boldsymbol{r}}\left[\frac{1}{T_p}\sum_{t=1}^{T_p} \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) - \ell(f^{\boldsymbol{b}}(X_t), Y_t^{\boldsymbol{b}})\right] \geq \frac{\epsilon}{16}.$$

Because this holds for any $p \geq 1$ and as $p \to \infty$ we have $T_p \to \infty$, we can now use Fatou lemma which yields

$$\mathbb{E}_{\boldsymbol{b},\boldsymbol{r}}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) - \ell(f^{\boldsymbol{b}}(X_t), Y_t^{\boldsymbol{b}})\right] \geq \frac{\epsilon}{16}.$$

This holds for any realization in $\mathcal{A} \cap \bigcap_{p\geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)$ which we recall has probability at least $\frac{\epsilon}{4}$. Therefore we finally obtain

$$\mathbb{E}_{\boldsymbol{b},\boldsymbol{r},\mathbb{X}}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t^{\boldsymbol{b}}) - \ell(f^{\boldsymbol{b}}(X_t), Y_t^{\boldsymbol{b}})\right] \geq \frac{\epsilon^2}{2^6}.$$

As a result, there exists a specific realization of $\boldsymbol{b}$ which we denote b such that

$$\mathbb{E}_{\boldsymbol{r},\mathbb{X}}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t^{b}) - \ell(f^{b}(X_t), Y_t^{b})\right] \geq \frac{\epsilon^2}{2^6},$$

which shows that with nonzero probability $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t^{b}) - \ell(f^{b}(X_t), Y_t^{b}) > 0$. This ends the proof of the theorem. As a remark, one can note that the construction of our bad example $\mathbb{Y}^b$ is a deterministic function of $\mathbb{X}$: it is independent from the realizations of the randomness used by the learning rule.

4.10.2 Proof of Lemma 4.2

We first construct our online learning algorithm, which is a simple variant of the classical exponential forecaster. We first define a step $\eta := \sqrt{2\ln t_0/t_0}$. At time $t = 1$ we always predict 0. For time step $t \geq 2$, we define the set $S_{t-1} := \{y \in \mathbb{N}, \sum_{u=1}^{t-1} \mathbb{1}_{y=y_u} > 0\}$ the set of values which have been visited. Then, we construct weights for all $y \in \mathbb{N}$ such that

$$w_{y,t-1} = \begin{cases} e^{\eta \sum_{u=1}^{t-1} \mathbb{1}_{y=y_u}}, & y \in S_{t-1} \\ 0 & \text{otherwise,} \end{cases}$$

and output a randomized prediction independent of the past history such that

$$\mathbb{P}(\hat{y}_t = y) = \frac{w_{y,t-1}}{\sum_{y'\in\mathbb{N}} w_{y',t-1}}.$$

This defines a proper online learning rule. Note that the denominator is well defined since $w_{y,t-1}$ is non-zero only for values in S_{t-1}, which contains at most $t-1$ elements. We now

define the expected success at time $1 \leq t \leq T$ as $\hat{s}_t := \frac{w_{y_t,t-1}}{\sum_{y\in\mathbb{N}} w_{y,t-1}} \mathbb{1}_{y_t \in S_t}$. Note that $\hat{s}_t = \mathbb{E}[\mathbb{1}_{f_t(\boldsymbol{y}_{\leq t-1})=y_t}]$. We first show that we have

$$\sum_{t=1}^{T} \hat{s}_t \geq \max_{y\in\mathbb{N}} \sum_{t=1}^{T} \mathbb{1}_{y=y_t} - \sqrt{T}\ln T.$$

To do so, we analyze the quantity $W_t := \frac{1}{\eta}\ln\left(\sum_{y\in S_t} e^{\eta\sum_{u=1}^{t}(\mathbb{1}_{y=y_u}-\hat{s}_u)}\right)$. Let $2 \leq t \leq T$. Supposing that $y_t \in S_{t-1}$, i.e., $S_t = S_{t-1}$, we define the operator $\Phi : \boldsymbol{x} \in \mathbb{R}^{|S_{t-1}|} \mapsto \frac{1}{\eta}\ln\left(\sum_{y\in S_{t-1}} e^{\eta x_y}\right)$ and use the Taylor expansion of Φ to obtain

$$\begin{aligned}
W_t &= \frac{1}{\eta}\ln\left(\sum_{y\in S_{t-1}} e^{\eta\sum_{u=1}^{t-1}(\mathbb{1}_{y=y_u}-\hat{s}_u)+\eta(\mathbb{1}_{y=y_t}-\hat{s}_t)}\right) \\
&= W_{t-1} + \sum_{y\in S_{t-1}} (\mathbb{1}_{y=y_t}-\hat{s}_t)\frac{e^{\eta\sum_{u=1}^{t-1}\mathbb{1}_{y=y_u}}}{\sum_{y'\in S_{t-1}} e^{\eta\sum_{u=1}^{t-1}\mathbb{1}_{y'=y_u}}} \\
&\qquad + \frac{1}{2}\sum_{y_1,y_2\in S_{t-1}} \left.\frac{\partial^2\Phi}{\partial x_{y_1}\partial x_{y_2}}\right|_{\xi} (\mathbb{1}_{y_1=y_u}-\hat{s}_u)(\mathbb{1}_{y_2=y_u}-\hat{s}_u) \\
&= W_{t-1} + \frac{1}{2}\sum_{y_1,y_2\in S_{t-1}} \left.\frac{\partial^2\Phi}{\partial x_{y_1}\partial x_{y_2}}\right|_{\xi} (\mathbb{1}_{y_1=y_t}-\hat{s}_u)(\mathbb{1}_{y_2=y_t}-\hat{s}_u) \\
&\leq W_{t-1} + \frac{1}{2}\sum_{y\in S_{t-1}} \frac{\eta e^{\eta\xi_y}}{\sum_{y'\in S_{t-1}} e^{\eta\xi_{y'}}} (\mathbb{1}_{y=y_t}-\hat{s}_u)^2 \\
&\leq W_{t-1} + \frac{\eta}{2},
\end{aligned}$$

for some vector $\xi \in \mathbb{R}^{|S_{t-1}|}$, where in the last inequality we used the fact $|\mathbb{1}_{y=y_t}-\hat{s}_u| \leq 1$. We now suppose that $y_t \notin S_{t-1}$ and $W_{t-1} \geq 1 + \frac{\ln 2 + 2\ln\frac{1}{\eta}}{\eta}$. In that case, $e^{\eta W_t} = e^{\eta W_{t-1}} + e^{\eta(1-\sum_{u=1}^{t-1}\hat{s}_u)}$. Hence, we obtain

$$W_t = W_{t-1} + \frac{\ln\left(1+e^{\eta(1-W_{t-1}-\sum_{u=1}^{t-1}\hat{s}_u)}\right)}{\eta} \leq W_{t-1} + \frac{e^{\eta(1-W_{t-1})}}{\eta} \leq W_{t-1} + \frac{\eta}{2}.$$

Now let $l = \max\{1\} \cup \left\{1 \leq t \leq T : W_t < 1 + \frac{\ln 2 + 2\ln\frac{1}{\eta}}{\eta}\right\}$. Note that for any $l < t \leq T$ the above arguments yield $W_t \leq W_{t-1} + \frac{\eta}{2}$. As a result, noting that $W_1 \leq 1$, we finally obtain

$$W_T \leq W_l + \eta\frac{T-l}{2} \leq 1 + \frac{\ln 2 + 2\ln\frac{1}{\eta}}{\eta} + \eta\frac{T}{2} \leq 1 + \ln 2\sqrt{\frac{t_0}{2\ln t_0}} + \sqrt{\frac{\ln t_0}{2t_0}}(t_0+T).$$

Therefore, for any $y \in S_T$, we have

$$\sum_{t=1}^{T}(\mathbb{1}_{y=y_t}-\hat{s}_t) \leq W_T \leq 1 + \ln 2\sqrt{\frac{t_0}{2\ln t_0}} + \sqrt{\frac{\ln t_0}{2t_0}}(t_0+T).$$

In particular, this shows that

$$\sum_{t=1}^{T} \hat{s}_t \geq \max_{y \in S_T} \sum_{t=1}^{T} \mathbb{1}_{y=y_t} - 1 - \ln 2 \sqrt{\frac{t_0}{2 \ln t_0}} - \sqrt{\frac{\ln t_0}{2t_0}}(t_0 + T).$$

Now note that if $y \notin S_T$, then $\sum_{t=1}^{T} \mathbb{1}_{y=y_t} = 0$, which yields

$$\max_{y \in S_T} \sum_{t=1}^{T} \mathbb{1}_{y=y_t} = \max_{y \in \mathbb{N}} \sum_{t=1}^{T} \mathbb{1}_{y=y_t}.$$

For the sake of conciseness, we will now denote by $\hat{y}_t$ the prediction of the online learning rule at time t. We observe that the variables $\mathbb{1}_{\hat{y}_t=y_t} - \hat{s}_t$ for $1 \leq t \leq T$ form a sequence of martingale differences. Further, $|\mathbb{1}_{\hat{y}_t=y_t} - \hat{s}_t| \leq 1$. Therefore, the Hoeffding-Azuma inequality shows that with probability $1 - \delta$,

$$\sum_{t=1}^{T} (\mathbb{1}_{\hat{y}_t=y_t} - \hat{s}_t) \geq -\sqrt{2T \ln \frac{1}{\delta}}.$$

Putting everything together yields that with probability $1 - \delta$,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{1}_{\hat{y}_t=y_t} &\geq \sum_{t=1}^{T} \hat{s}_t - \sqrt{2T \ln \frac{1}{\delta}} \\
&\geq \max_{y \in \mathbb{N}} \sum_{t=1}^{T} \mathbb{1}_{y=y_t} - 1 - \ln 2 \sqrt{\frac{t_0}{2 \ln t_0}} - \sqrt{\frac{\ln t_0}{2t_0}}(t_0 + T) - \sqrt{2T \ln \frac{1}{\delta}}.
\end{aligned}$$

This ends the proof of the lemma.

4.10.3 Proof of Theorem 4.9

We use a similar learning rule to the one constructed in Section 4.4 for totally-bounded spaces. We only make a slight modification of the learning rules $f_\cdot^\epsilon$. Precisely, we pose for $0 < \epsilon \leq 1$,

$$T_\epsilon := \left\lceil \frac{2^4 \cdot 3^2 (1 + \ln \frac{1}{\epsilon})}{\epsilon^2} \right\rceil \quad \text{and} \quad \delta_\epsilon := \frac{\epsilon}{2T_\epsilon}.$$

Then, let ϕ be the representative function from the $(1 + \delta_\epsilon)$C1NN learning rule. Similarly as for the $\epsilon-$approximation learning rule from Section 4.4, we consider the same equivalence relation $\overset{\phi}{\sim}$ on times to define clusters. The learning rule then performs its prediction based on the values observed on the corresponding cluster using the learning rule from Lemma 4.2 using $t_0 = T_\epsilon$. Precisely, let $\eta_\epsilon := \sqrt{2 \ln T_\epsilon / T_\epsilon}$ and define the weights $w_{y,t} = e^{\eta_\epsilon \sum_{u<t : u \overset{\phi}{\sim} t} \mathbb{1}(Y_u = y)}$ for all $y \in \tilde{S} := \{y' \in \mathbb{N} : \sum_{u<t : u \overset{\phi}{\sim} t} \mathbb{1}(Y_u = y') > 0\}$ and $w_{y,t} = 0$ otherwise. The learning rule $f_t^\epsilon(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t)$ outputs a random value in $\mathbb{N}$ independent of the past history such that

$$\mathbb{P}(\hat{Y}_t = y) = \frac{w_{y,t}}{\sum_{y' \in \mathbb{N}} w_{y',t}}, \quad y \in \mathbb{N}.$$

The final learning rule $f_.$ is then defined similarly as before from the learning rules $f_.^\epsilon$ for $\epsilon > 0$. Therefore, Lemma 4.1 still holds. Also, using the same notations as in the proof of Theorem 4.6, Lemma 4.2 implies that for any $t \geq 1$, we can write for any $t \geq 1$ on the cluster $\mathcal{C}(t) = \{u < t : u \overset{\phi}{\sim} t\}$,

$$\begin{aligned}
\sum_{u \in \mathcal{C}(t)} \bar{\ell}_{01}(\hat{Y}_u(\epsilon), Y_u) &\leq \min_{y \in \mathbb{N}} \sum_{u \in \mathcal{C}(t)} \ell_{01}(y, Y_u) + 1 + \ln 2 \sqrt{\frac{T_\epsilon}{2 \ln T_\epsilon}} + \sqrt{\frac{\ln T_\epsilon}{2T_\epsilon}} (T_\epsilon + |\mathcal{C}(t)|) \\
&\leq \min_{y \in \mathbb{N}} \sum_{u \in \mathcal{C}(t)} \ell_{01}(y, Y_u) + \left(\frac{1}{T_\epsilon} + \frac{\ln 2}{\sqrt{2T_\epsilon \ln T_\epsilon}} + \sqrt{\frac{2 \ln T_\epsilon}{T_\epsilon}} \right) \max(T_\epsilon, |\mathcal{C}(t)|) \\
&\leq \min_{y \in \mathbb{N}} \sum_{u \in \mathcal{C}(t)} \ell_{01}(y, Y_u) + \left(\frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \right) \max(T_\epsilon, |\mathcal{C}(t)|) \\
&= \min_{y \in \mathbb{N}} \sum_{u \in \mathcal{C}(t)} \ell_{01}(y, Y_u) + \epsilon \max(T_\epsilon, |\mathcal{C}(t)|)
\end{aligned}$$

Therefore, the same proof of Theorem 4.6 holds by replacing all $\epsilon-$nets $\mathcal{Y}_\epsilon$ directly by $\mathbb{N}$. The martingale argument still holds since the learning rule used is indeed online. This ends the proof of this theorem.

4.10.4 Proof of Theorem 4.10

We first need the following simple result which intuitively shows that we can assume that the predictions of the learning rule for mean estimation $g^\epsilon_{\leq t_\epsilon}$ are unrelated for $t = 1, \dots, t_\epsilon$.

Lemma 4.7. *Let $(\mathcal{Y}, \ell)$ satisfying F-TiME. For any $\eta > 0$, there exists a horizon time $T_\eta \geq 1$, an online learning rule $g_{\leq T_\eta}$ such that for any $\boldsymbol{y} := (y_t)_{t=1}^{T_\eta}$ of values in $\mathcal{Y}$ and any value $y \in \mathcal{Y}$, we have*

$$\frac{1}{T_\eta} \mathbb{E} \left[\sum_{t=1}^{T_\eta} \ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t) \right] \leq \eta,$$

and such that the random variables $g_t(\boldsymbol{y}_{\leq t-1})$ are independent.

Proof Fix $\eta > 0$, $T_\eta \geq 1$ and $g_{\leq T_\eta}$ such that this online learning rule satisfies the condition from F-TiME for $\eta > 0$. We consider the following learning rule $\tilde{g}_.$. For any $t \geq 1$ and $\boldsymbol{y} \in \mathcal{Y}^{t-1}$,

$$\tilde{g}_t(\boldsymbol{y}_{\leq t-1}) = g_t^t(\boldsymbol{y}_{\leq t-1}),$$

where $(g_.^t)$ are i.i.d. samples of the learning rule $g_.$. By construction, we still have that for any sequence $\boldsymbol{y}_{T_\eta} \in \mathcal{Y}^{T_\eta}$,

$$\frac{1}{T_\eta} \mathbb{E} \left[\sum_{t=1}^{T_\eta} \ell(\tilde{g}_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t) \right] = \frac{1}{T_\eta} \mathbb{E} \left[\sum_{t=1}^{T_\eta} \ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t) \right] \leq \eta.$$

This ends the proof of the lemma. ■

From now on, by Lemma 4.7, we will suppose without loss of generality that the learning rule g^ϵ has predictions that are independent at each step (conditionally on the observed values). For simplicity, we refer to the prediction of the defined learning rule $f_\cdot$ (resp. $f_\cdot^\epsilon$) at time t as $\hat{Y}_t$ (resp. $\hat{Y}_t(\epsilon)$). We now show that is optimistically universal for arbitrary responses. By construction of the learning rule $f_\cdot$, Lemma 4.1 still holds. Therefore, we only have to focus on the learning rules $f_\cdot^\epsilon$ and prove that we obtain similar results as before. Let $T \geq 1$ and denote by $\mathcal{A}_i := \{t \leq T : |\{u \leq T : \phi(u) = t\}| = i\}$ the set of times which have exactly i children within horizon T for $i = 0, 1, 2$. Then, we define

$$\mathcal{B}_T = \{t \leq T : L_t = 0 \text{ and } |\{t < u \leq T : u \overset{\phi}{\sim} t\}| \geq t_\epsilon\},$$

i.e., times that start a new learning block and such that there are at least t_ϵ future times falling in their cluster within horizon T. Note that the function ψ defines a parent-relation (similarly to ϕ, but defined for all times $t \geq 1$). To simplify notations, for any $t \in \mathcal{B}_T$, we denote t^u the $\psi-$children of t at generation $u-1$ for $1 \leq u \leq t_\epsilon$, i.e., we have $\psi^{u-1}(t^u) = t$ for all $1 \leq u \leq t_\epsilon$. In particular $t = t^1$. By construction, blocks have length at most t_ϵ. More precisely, the block started at any $t \in \mathcal{B}_T$ has had time to finish completely, hence has length exactly t_ϵ. By construction of the indices L_t, the blocks $\{t^u, 1 \leq u \leq t_\epsilon\}$, for $t \in \mathcal{B}_T$, are all disjoint. This implies in particular $|\mathcal{B}_T| t_\epsilon \leq T$. We first analyze the predictions along these blocks and for any $t \in \mathcal{B}_T$ and $y \in \mathcal{Y}$, we pose $\delta_t(y) := \frac{1}{t_\epsilon} \sum_{u=1}^{t_\epsilon} \Big(\ell(\hat{Y}_{t^u}, Y_{t^u}) - \ell(y, Y_{t^u}) - \epsilon \Big)$. Now by construction of the learning rule $f_\cdot^\epsilon$, we have

$$t_\epsilon \delta_t(y^t) = \sum_{u=1}^{t_\epsilon} \big(\ell(g_u^{\epsilon,t}(\{Y_{t^l}\}_{l=1}^{u-1}), Y_{t^u}) - \ell(y^t, Y_{t^u}) \big) - \epsilon t_\epsilon.$$

Next, for any $t \leq t_\epsilon$ and sequence $\boldsymbol{y}_{\leq t-1}$ and value $y \in \mathcal{Y}$, we write $\bar{\ell}(g_t^\epsilon(\boldsymbol{y}_{\leq t-1}), y) := \mathbb{E}\left[\ell(g_t^\epsilon(\boldsymbol{y}_{\leq t-1}), y)\right]$. Now by hypothesis on the learning rule $g_{\leq t_\epsilon}^\epsilon$,

$$\frac{1}{t_\epsilon} \sum_{u=1}^{t_\epsilon} \bar{\ell}(\hat{Y}_{t^u}, Y_{t^u}) - \ell(y^t, Y_{t^u}) \leq \epsilon. \tag{4.2}$$

Now consider the following sequence $(\ell(\hat{Y}_{t^u}, Y_{t^u}) - \bar{\ell}(\hat{Y}_{t^u}, Y_{t^u}))_{t \in \mathcal{B}_T, 1 \leq u \leq s(t)}$. Because of the definition of the learning rule, which uses i.i.d. copies of the learning rule $g_\cdot^\epsilon$, if we order the former sequence by increasing order of t^u, we obtain a sequence of martingale differences. We can continue this sequence by zeros to ensure that it has length exactly T. As a result, we obtain a sequence of T martingale differences, which are all bounded by $\bar{\ell}$ in absolute value. Now, the Azuma-Hoeffding inequality implies that for $\delta > 0$, with probability $1 - \delta$, we have

$$\sum_{t \in \mathcal{B}_T} \sum_{u=1}^{t_\epsilon} \ell(\hat{Y}_{t^u}, Y_{t^u}) \leq \sum_{t \in \mathcal{B}_T} \sum_{u=1}^{t_\epsilon} \bar{\ell}(\hat{Y}_{t^u}, Y_{t^u}) + \bar{\ell} \sqrt{2T \ln \frac{1}{\delta}}.$$

Thus, using Eq (4.2), with probability at least $1 - \delta$,

$$\sum_{t \in \mathcal{B}_T} t_\epsilon \delta_t(y^t) \leq \bar{\ell} \sqrt{2T \ln \frac{1}{\delta}}. \tag{4.3}$$

We also denote $\mathcal{T} = \bigcup_{t\in\mathcal{B}_T}\{t^u, 1 \le u \le t_\epsilon\}$ the union of all blocks within horizon T. This set contains all times $t \le T$ except *bad* times close to the last times of their corresponding cluster $\{u \le T : u \overset{\phi}{\sim} t\}$. Precisely, these are times t such that $|\{t < u \le T : u \overset{\phi}{\sim} t\}| < t_\epsilon - L_t$. As a result, there are at most t_ϵ such times for each cluster. Using the same arguments as in the proof of Theorem 4.6, if we consider only clusters of duplicates (i.e., the cluster started for a specific instance which has high number of duplicates), the corresponding *bad* times contribute to a proportion $\le \frac{t_\epsilon}{T_\epsilon/\epsilon} \le \epsilon^2$ of times. Now consider clusters that have at least T_ϵ times. Their *bad* times contribute to a proportion $\le \frac{t_\epsilon}{T_\epsilon} \le \epsilon$ of times. Last, we need to account for clusters of size $< T_\epsilon$ which necessarily contain leaves of the tree ϕ: there are at most $|\mathcal{A}_0|$ such clusters. By the Chernoff bound, with probability at least $1 - e^{-T\delta_\epsilon/3}$ we have

$$T - |\mathcal{T}| \le (\epsilon^2 + \epsilon)T + |\mathcal{A}_0|t_\epsilon \le t_\epsilon + (\epsilon^2 + \epsilon + 2\delta_\epsilon t_\epsilon)T \le t_\epsilon + 3\epsilon T.$$

By the Borel-Cantelli lemma, because $\sum_{T\ge1} e^{-T\delta_\epsilon/3} < \infty$, almost surely there exists a time $\hat{T}$ such that for $T \ge \hat{T}$ we have $T - |\mathcal{T}| \le t_\epsilon + 3\epsilon T$. We denote by $\mathcal{E}_\epsilon$ this event. Then, on the event $\mathcal{E}_\epsilon$, for any $T \ge \hat{T}$ and for any sequence of values $(y^t)_{t\ge1}$ we have

$$\begin{aligned}
\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon), Y_t) &\le \sum_{t\in\mathcal{B}_T}\sum_{u=1}^{t_\epsilon} \ell(\hat{Y}_{t^u}, Y_{t^u}) + (T - |\mathcal{T}|)\bar{\ell} \\
&\le \sum_{t\in\mathcal{B}_T}\sum_{u=1}^{t_\epsilon} \ell(y^t, Y_{t^u}) + \sum_{t\in\mathcal{B}_T} t_\epsilon\delta_t(y^t) + \epsilon|\mathcal{B}_T|t_\epsilon + t_\epsilon\bar{\ell} + 3\epsilon T \\
&\le \sum_{t\in\mathcal{B}_T}\sum_{u=1}^{t_\epsilon} \ell(y^t, Y_{t^u}) + \sum_{t\in\mathcal{B}_T} t_\epsilon\delta_t(y^t) + t_\epsilon\bar{\ell} + 4\epsilon T.
\end{aligned}$$

Now let $f : \mathcal{X} \to \mathcal{Y}$ be a measurable function to which we compare $f^\epsilon_\cdot$. By Theorem 4.5, because $(1+\delta_\epsilon)$C1NN is optimistically universal without noise and $\mathbb{X} \in$ Soul, almost surely $\frac{1}{T}\sum_{t=1}^{T} \ell(f(X_{\phi(t)}), f(X_t)) \to 0$. We denote by $\mathcal{F}_\epsilon$ this event of probability one. The proof of Theorem 4.6 shows that on $\mathcal{F}_\epsilon$, for any $0 \le u \le T_\epsilon - 1$ we have

$$\frac{1}{T}\sum_{t=1}^{T} \ell(f(X_{\phi^u(t)}), f(X_t)) \to 0.$$

We let $y^t = f(X_t)$ for all $t \ge 1$. Then, recalling that for any $t \in \mathcal{B}_T$, we have $t = \phi^{u-1}(t^u)$, on the event $\mathcal{E}_\epsilon$, for any $T \ge \hat{T}$ we have

$$\begin{aligned}
&\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon), Y_t) \\
&\le \sum_{t\in\mathcal{B}_T}\sum_{u=1}^{t_\epsilon} \big((1+\epsilon)\ell(f(X_{t^u}), Y_{t^u}) + c^\ell_\epsilon \ell(f(X_t), f(X_{t^u}))\big) + \sum_{t\in\mathcal{B}_T} t_\epsilon\delta_t(y^t) + t_\epsilon\bar{\ell} + 4\epsilon T \\
&\le \sum_{t=1}^{T} \ell(f(X_t), Y_t) + c^\ell_\epsilon\frac{T_\epsilon}{\epsilon}\sum_{u=0}^{T_\epsilon-1}\sum_{t=1}^{T} \ell(f(X_{\phi^u(t)}), f(X_t)) + \sum_{t\in\mathcal{B}_T} t_\epsilon\delta_{\varphi(t)}(y^t) + t_\epsilon\bar{\ell} + 5\epsilon T,
\end{aligned}$$

where in the first inequality we used the generalized-metric loss identity, and in the second inequality we used the fact that cluster with distinct instance values have at most $\frac{T_\epsilon}{\epsilon}$ duplicates of each instance. Next, using Eq (4.3), with probability $1 - \frac{1}{T^2}$, we have

$$\sum_{t \in \mathcal{B}_T} t_\epsilon \delta_t(y^t) \leq 2\bar{\ell}\sqrt{T \ln T}.$$

Because $\sum_{T \geq 1} \frac{1}{T^2} < 0$, the Borel-Cantelli lemma implies that on an event $\mathcal{G}_\epsilon$ of probability one, there exists $\hat{T}_2$ such that for all $T \geq \hat{T}_2$ the above inequality holds. As a result, on the event $\mathcal{E}_\epsilon \cap \mathcal{F}_\epsilon \cap \mathcal{G}_\epsilon$ we obtain for any $T \geq \max(\hat{T}, \hat{T}_2)$ that

$$\sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon), Y_t) \leq \sum_{t=1}^{T} \ell(f(X_t), Y_t) + \frac{c_\epsilon^\ell T_\epsilon}{\epsilon} \sum_{u=0}^{T_\epsilon - 1} \sum_{t=1}^{T} \ell(f(X_{\phi^u(t)}), f(X_t)) \\ + 2\bar{\ell}\sqrt{T \ln T} + t_\epsilon \bar{\ell} + 5\epsilon T.$$

where $\frac{1}{T} \sum_{u=0}^{T_\epsilon - 1} \sum_{t=1}^{T} \ell(f(X_{\phi^u(t)}), f(X_t)) \to 0$ because the event $\mathcal{F}_\epsilon$ is met. Therefore, we obtain that on the event $\mathcal{E}_\epsilon \cap \mathcal{F}_\epsilon \cap \mathcal{G}_\epsilon$ of probability one,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \Big[\ell(\hat{Y}_t(\epsilon), Y_t) - \ell(f(X_t), Y_t) \Big] \leq 5\epsilon,$$

i.e., almost surely, the learning rule $f_\cdot^\epsilon$ achieves risk at most 5ϵ compared to the fixed function f. By union bound, on the event $\bigcap_{i \geq 0} (\mathcal{E}_{\epsilon_i} \cap \mathcal{F}_{\epsilon_i} \cap \mathcal{G}_{\epsilon_i})$ of probability one we have that

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \Big[\ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) \Big] \leq 5\epsilon_i, \quad \forall i \geq 0.$$

The rest of the proof uses similar arguments as in the proof of Theorem 4.6. Precisely, let $\mathcal{H}$ be the almost sure event of Lemma 4.1 such that there exists $\hat{t}$ for which

$$\forall t \geq \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^{t} \ell(\hat{Y}_t, Y_t) \leq \sum_{s=t_i}^{t} \ell(\hat{Y}_t(\epsilon_i), Y_t) + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{t \ln t}.$$

In the rest of the proof we will suppose that the event $\mathcal{H} \cap \bigcap_{i \geq 0} (\mathcal{E}_{\epsilon_i} \cap \mathcal{F}_{\epsilon_i} \cap \mathcal{G}_{\epsilon_i})$ of probability one is met. Let $i \geq 0$. For all $t \geq \max(\hat{t}, t_i)$ we have

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) &\leq \frac{t_i}{T}\bar{\ell} + \frac{1}{T} \sum_{t=t_i}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \\
&\leq \frac{t_i}{T}\bar{\ell} + \frac{1}{T} \sum_{t=t_i}^{T} \ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{\frac{\ln T}{T}} \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t(\epsilon_i), Y_t) - \ell(f(X_t), Y_t) + \frac{2t_i}{T}\bar{\ell} + (2 + \bar{\ell} + \bar{\ell}^2)\sqrt{\frac{\ln T}{T}}.
\end{aligned}$$

Therefore we obtain $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 5\epsilon_i$. Because this holds for any $i \geq 0$ we finally obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 0.$$

As a result, $f.$ is universally consistent for adversarial responses under all SOUL processes. Hence, SOLAR = SOUL and $f.$ is in fact optimistically universal. This ends the proof of the theorem.

4.10.5 Proof of Lemma 4.3

We first note that with the same horizon time T_η, we have that F-TiME implies Property 2. We now show that Property 2 implies F-TiME. Let $(\mathcal{Y}, \ell)$ satisfying Property 2. We now fix $\eta > 0$ and let $T, g_{\leq\tau}$ such that for any $\boldsymbol{y} := (y_t)_{t=1}^{T}$ of values in $\mathcal{Y}$ and any value $y \in \mathcal{Y}$, we have

$$\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{\tau} \left(\ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)\right)\right] \leq \eta.$$

We now construct a random time $1 \leq \tilde{\tau} \leq T$ such that $\mathbb{P}[\tilde{\tau} = t] = \frac{\mathbb{P}[\tau=t]}{t\mathbb{E}[1/\tau]}$ for all $1 \leq t \leq T$. This indeed defines a proper random variable because $\sum_{t=1}^{T} \frac{\mathbb{P}[\tau=t]}{t\mathbb{E}[1/\tau]} = 1$. Let $Supp(\tau) := \{1 \leq t \leq T : \mathbb{P}[\tau = t] > 0\}$ be the support of τ. For any $t \in Supp(\tau)$, we denote by $g^t_{\leq t}$ the learning rule obtained by conditioning $g_{\leq\tau}$ on the event $\{\tau = t\}$, i.e., $g^t_{\leq t} = g_{\leq\tau}|\tau = t$. More precisely, recall that τ only uses the randomness of g_t. It is not an online random time. Hence, a practical way to simulate $g^t_{\leq t}$ for all $t \in Supp(\tau)$ is to first draw an i.i.d. sequence of learning rules $(g_{i,\leq\tau_i})_{i\geq 1}$. Then, for each $t \in Supp(\tau)$ we select the randomness which first satisfies $\tau = t$. Specifically, we define the time $i_t = \min\{i : \tau_i = t\}$ for all $t \in Supp(\tau)$. With probability one, these times are finite for all $t \in Supp(\tau)$. Denote this event $\mathcal{E}$. Then, letting $\bar{y} \in \mathcal{Y}$ be an arbitrary fixed value, for all $1 \leq t \leq T$ we pose

$$g^t_{\leq t} = \begin{cases} g_{i_t,\leq t} & \text{if } \mathcal{E} \text{ is met,} \\ \bar{y}_{\leq t} & \text{otherwise,} \end{cases} \quad t \in Supp(\tau) \quad \text{and} \quad g^t_{\leq t} = \bar{y}_{\leq t}, \quad t \notin Supp(\tau).$$

where $\bar{y}_{\leq t}$ denotes the learning rules which always outputs value $\bar{y}$ for all steps $u \leq t$. Intuitively, $g^t_{\leq t}$ has the same distribution as $g_{\leq\tau}$ conditioned on the event $\{\tau = t\}$. We are now ready to define a new learning rule $\tilde{g}_{\leq\tilde{\tau}}$, by $\tilde{g}_{\leq\tilde{\tau}} := g^{\tilde{\tau}}_{\leq\tilde{\tau}}$. Noting that for any $t \notin Supp(\tau)$

we have $\mathbb{P}[\tilde{\tau} = t] = 0$, we can write

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{\tau}(\ell(\tilde{g}_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)) - \eta\tau\right] \\
&= \sum_{t=1}^{T}\mathbb{P}[\tilde{\tau} = t]\mathbb{E}\left[\sum_{u=1}^{t}(\ell(\tilde{g}_u(\boldsymbol{y}_{\leq u-1}), y_u) - \ell(y, y_u)) - \eta t \middle| \tilde{\tau} = t\right] \\
&= \sum_{t\in Supp(\tau)}\mathbb{P}[\tilde{\tau} = t]\mathbb{E}\left[\sum_{u=1}^{t}(\ell(\tilde{g}_u(\boldsymbol{y}_{\leq u-1}), y_u) - \ell(y, y_u)) - \eta t \middle| \tilde{\tau} = t, \mathcal{E}\right] \\
&= \frac{1}{\mathbb{E}[1/\tau]}\sum_{t\in Supp(\tau)}\mathbb{P}[\tau = t]\mathbb{E}\left[\frac{1}{t}\sum_{u=1}^{t}(\ell(g_{i_t,u}(\boldsymbol{y}_{\leq u-1}), y_u) - \ell(y, y_u)) - \eta \middle| \tilde{\tau} = t, \mathcal{E}\right] \\
&= \frac{1}{\mathbb{E}[1/\tau]}\sum_{t\in Supp(\tau)}\mathbb{P}[\tau = t]\mathbb{E}\left[\frac{1}{t}\sum_{u=1}^{t}(\ell(g_{i_t,u}(\boldsymbol{y}_{\leq u-1}), y_u) - \ell(y, y_u)) - \eta\right] \\
&= \frac{1}{\mathbb{E}[1/\tau]}\sum_{t\in Supp(\tau)}\mathbb{P}[\tau = t]\mathbb{E}\left[\frac{1}{t}\sum_{u=1}^{t}(\ell(g_u(\boldsymbol{y}_{\leq u-1}), y_u) - \ell(y, y_u)) - \eta \middle| \tau = t\right] \\
&= \frac{1}{\mathbb{E}[1/\tau]}\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{\tau}(\ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)) - \eta\right] \leq 0.
\end{aligned}$$

where in the second and fourth equality we used the fact that $\mathbb{P}[\mathcal{E}] = 1$. As a result, there exists a learning rule $\tilde{g}_{\leq\tilde{\tau}}$ such that $1 \leq \tilde{\tau} \leq T_\eta$, and for any $\boldsymbol{y}_{\leq T_\eta} \in \mathcal{Y}^{T_\eta}$ and $y \in \mathcal{Y}$ one has

$$\mathbb{E}\left[\sum_{t=1}^{\tilde{\tau}}(\ell(\tilde{g}_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)) - \eta\tilde{\tau}\right] \leq 0.$$

We now pose $T'_\eta = \lceil T_\eta/\eta \rceil$ and draw an i.i.d. sequence of learning rules $(\tilde{g}^i_{\leq\tilde{\tau}_i})_{i\geq 1}$. Denote $\theta_i = \sum_{j<i}\tilde{\tau}_i$ with the convention $\theta_1 = 0$. We are now ready to define a learning rule $h_{\leq T'_\eta}$ as follows. For any $1 \leq t \leq T'_\eta$ and $\boldsymbol{y}_{\leq t} \in \mathcal{Y}^t$,

$$h_t(\boldsymbol{y}_{\leq t-1}) = \tilde{g}^i_{\leq t-\theta_i}((y_{t'})_{\theta_i<t'\leq t-1}), \qquad \theta_i < t \leq \theta_{i+1}, i \geq 1.$$

In other words, the learning rule performs independent learning rules $\tilde{g}_{\leq\tilde{\tau}}$ and when the time horizon $\tilde{\tau}$ is reached, we re-initialize the learning rule with a new randomness. Now let $\boldsymbol{y}_{\leq T'_\eta} \in \mathcal{Y}^{T'_\eta}$ and $y \in \mathcal{Y}$. We denote by $\hat{i} = \max\{i \geq 1, \theta_i \leq t\}$, the index of the last learning

rule which had time to finish completely. Then, because $\tilde{\tau}_{\hat{i}} \leq T_\eta$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T'_\eta}(\ell(h_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t)) - 2\eta T'_\eta\right] \\
&\leq \mathbb{E}\left[\sum_{i\leq\hat{i}}\sum_{t=1}^{\tilde{\tau}_i}(\ell(\tilde{g}^i_{t-\theta_i}(\boldsymbol{y}_{\theta_i<\cdot\leq t-1}), y_t) - \ell(y, y_t)) - \eta T'_\eta\right] - \eta T'_\eta + T_\eta \\
&\leq \mathbb{E}\left[\sum_{i\leq\hat{i}}\left(\sum_{t=1}^{\tilde{\tau}_i}(\ell(\tilde{g}^i_{t-\theta_i}(\boldsymbol{y}_{\theta_i<\cdot\leq t-1}), y_t) - \ell(y, y_t)) - \eta\tilde{\tau}_i\right)\right].
\end{aligned}$$

We now analyze the last term. First, note that by construction, the sequence

$$\left\{S_j := \sum_{j\leq i}\left(\sum_{t=1}^{\tilde{\tau}_j}(\ell(\tilde{g}^j_{t-\theta_j}(\boldsymbol{y}_{\theta_j<\cdot\leq t-1}), y_t) - \ell(y, y_t)) - \eta\tilde{\tau}_j\right)\right\}_{j\geq 1}$$

is a super-martingale. Now, note that $\hat{i} \leq 1 + T'_\eta$ since for all i, $\theta_i = \sum_{j<i}\tau_i \geq i - 1$. As a result, $\hat{i}$ is bounded, is a stopping time for the considered filtration (after finishing period $\hat{i}$ we stop if and only we exceed time T'_η) and we can apply Doob's optimal sampling theorem to obtain $\mathbb{E}[S_{\hat{i}}] \leq 0$. Thus, combining the above equations gives

$$\frac{1}{T'_\eta}\mathbb{E}\left[\sum_{t=1}^{T'_\eta}(\ell(h_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t))\right] \leq 2\eta.$$

Because this holds for all $\eta > 0$, F-TiME is satisfied. This ends the proof of the lemma.

4.10.6 Proof of Lemma 4.11

We first prove that adversarial regression for processes outside of CS is not achievable. Precisely, we show that for any $\mathbb{X} \notin \mathrm{CS}$, for any online learning rule $f.$, there exists a process $\mathbb{Y}$ on $\mathcal{Y}$, a measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ and $\delta > 0$ such that with non-zero probability $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^*) > \delta$.

Because F-TiME is not satisfied by $(\mathcal{Y}, \ell)$, by Lemma 4.3, Property 2 is not satisfied either. Hence, we can fix $\eta > 0$ such that for any horizon $T \geq 1$ and any online learning rule $g_{\leq\tau}$ with $1 \leq \tau \leq T$, there exist a sequence $\boldsymbol{y} := (y_t)_{t=1}^T$ of values in $\mathcal{Y}$ and a value y such that

$$\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{\tau}(\ell(g_t(\boldsymbol{y}_{\leq t-1}), y_t) - \ell(y, y_t))\right] > \eta,$$

as in the assumption of the space $(\mathcal{Y}, \ell)$. Let $\mathbb{X} \notin \mathrm{CS}$. The proof of Theorem 4.7 shows that there exist $0 < \epsilon < 1$, a sequence of disjoint measurable sets $\{B_p\}_{p\geq 1}$ and a sequence of times $(t_p)_{p\geq 0}$ with $t_0 = 0$ and such that with $\mu := \max(1, \frac{8\bar{\ell}}{\epsilon\eta})$, for any $p \geq 1$, $t_p > \mu t_{p-1}$, and

defining the events

$$\mathcal{E}_p = \left\{ \mathbb{X}_{\leq t_{p-1}} \cap \left(\bigcup_{p' \geq p} B_p \right) = \emptyset \right\} \text{ and } \mathcal{F}_p := \bigcup_{\mu t_{p-1} < t \leq t_p} \left\{ \frac{1}{t} \sum_{u=1}^{t} \mathbb{1}_{B_p}(X_u) \geq \frac{\epsilon}{4} \right\},$$

we have $\mathbb{P}[\bigcap_{p\geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)] \geq \frac{\epsilon}{4}$. We now fix a learning rule $f.$ and construct a "bad" process $\mathbb{Y}$ recursively. Fix $\bar{y} \in \mathcal{Y}$ an arbitrary value. We start by defining the random variables $N_p(t) = \sum_{u=t_{p-1}+1}^{t} \mathbb{1}_{B_p}(X_u)$ for any $p \geq 1$. We now construct (deterministic) values y_p and sequences $(y_p^u)_{u=1}^{t_p}$ for all $p \geq 1$, of values in $\mathcal{Y}$. Suppose we have already constructed the values y_q as well as the sequences $(y_q^u)_{u=1}^{t_q}$ for all $q < p$. We will now construct y_p and $(y_p^u)_{u=1}^{t_p}$. Assuming that the event $\mathcal{E}_p \cap \mathcal{F}_p$ is met, there exists $\mu t_{p-1} < t \leq t_p$ such that

$$N_p(t) = \sum_{u=t_{p-1}+1}^{t} \mathbb{1}_{B_p}(X_u) = \sum_{u=1}^{t} \mathbb{1}_{B_p}(X_u) \geq \frac{\epsilon}{4} t,$$

where in the first equality we used the fact that on $\mathcal{E}_p$, the process $\mathbb{X}_{\leq t_{p-1}}$ does not visit B_p. In the rest of the construction, we will denote

$$T_p = \begin{cases} \min\{\mu t_{p-1} < t \leq t_p : N_p(t) \geq \frac{\epsilon}{4} t\} & \text{if } \mathcal{E}_p \cap \mathcal{F}_p \text{ is met.} \\ t_p & \text{otherwise.} \end{cases}$$

Now consider the process $\mathbb{Y}_{t \leq t_{p-1}}(\mathbb{X})$ defined as follows. For any $1 \leq q < p$ we pose

$$Y_t(\mathbb{X}) = \begin{cases} y_q^{N_q(t)} & \text{if } t \leq T_q \text{ and } X_t \in B_q, \\ y_q & \text{if } t > T_q \text{ and } X_t \in B_q, \\ y_{q'} & \text{if } X_t \in B_{q'},\ q' < q, \\ \bar{y} & \text{otherwise,} \end{cases} \qquad t_{q-1} < t \leq t_q.$$

Similarly, for $M \geq 1$ and given any sequence $\{\tilde{y}_i\}_{i=1}^{M}$, we define the following process $\mathbb{Y}_{t_{p-1}<u\leq t_p}\left(\mathbb{X}, \{\tilde{y}_i\}_{i=1}^{M}\right)$ by

$$Y_u\left(\mathbb{X}, \{\tilde{y}_i\}_{i=q1}^{M}\right) = \begin{cases} \tilde{y}_{\min(N_p(u),M)} & \text{if } X_t \in B_p, \\ y_q & \text{if } X_t \in B_q,\ q < p, \\ \bar{y} & \text{otherwise.} \end{cases}$$

We now construct a learning rule $g_\cdot^p$. First, we define the event $\mathcal{B} := \bigcap_{p\geq 1}(\mathcal{E}_p \cap \mathcal{F}_p)$. We will denote by $\tilde{\mathbb{X}} = \mathbb{X}|\mathcal{B}$ a sampling of the process $\mathbb{X}$ on the event $\mathcal{B}$ which has probability at least $\frac{\epsilon}{4}$. For instance we draw i.i.d. samplings following the same distribution as $\mathbb{X}$ then select the process which first falls into $\mathcal{B}$. We are now ready to define a learning rule $(g_u^p)_{u\leq\tau}$ where τ is a random time. To do so, we first draw a sample $\tilde{\mathbb{X}}$ which is now fixed for the learning rule $g_\cdot^p$. We define the stopping time as $\tau = N_p(T_p)$. Finally, for all $1 \leq u \leq \tau$, and any sequence of values $\tilde{\boldsymbol{y}}_{\leq u-1}$, we pose

$$g_u^p(\tilde{\boldsymbol{y}}_{\leq u-1}) = f_{T_p(u)}\left(\tilde{\mathbb{X}}_{\leq T_p(u)-1}, \left\{\mathbb{Y}_{\leq t_{p-1}}(\tilde{\mathbb{X}}), \mathbb{Y}_{t_{p-1}<u\leq T_p(u)-1}\left(\tilde{\mathbb{X}}, \{\tilde{y}_i\}_{i=1}^{u-1}\right)\right\}, \tilde{X}_{T_p(u)}\right),$$

where we used the notation $T_p(u) := \min\{t_{p-1} < t' \leq t_p : N_p(t) = u\}$ for the time of the $u-$th visit of B_p, which exists because $u \leq \tau = N_p(T_p) \leq N_p(t_p)$ since the event $\mathcal{B}$ is satisfied by $\tilde{\mathbb{X}}$. Note that the prediction of the rule $g^p_\cdot$ is random because of the dependence on $\tilde{\mathbb{X}}$. Also, observe that the random time τ is bounded by $1 \leq \tau \leq T_p \leq t_p$. Therefore, by hypothesis on the value space $(\mathcal{Y}, \ell)$, there exists a sequence $\{y_p^u\}_{u=1}^{t_p}$ and a value $y_p \in \mathcal{Y}$ such that

$$\mathbb{E}\left[\frac{1}{\tau}\sum_{u=1}^{\tau}\left(\ell(g_u^p(\boldsymbol{y}_p{}^{\leq u-1}), y_p^u) - \ell(y_p, y_p^u)\right)\right] \geq \eta.$$

This ends the recursive construction of the values y_p and the sequences $(y_p^u)_{u=1}^{t_p}$ for all $p \geq 1$. We are now ready to define the process $\mathbb{Y}(\mathbb{X})$, using a similar construction as before. For any $p \geq 1$ we define

$$Y_t(\mathbb{X}) = \begin{cases} y_p^{N_p(t)} & \text{if } t \leq T_p \text{ and } X_t \in B_p, \\ y_p & \text{if } t > T_p \text{ and } X_t \in B_p, \\ y_q & \text{if } X_t \in B_q,\ q < p, \\ \bar{y} & \text{otherwise,} \end{cases} \qquad t_{p-1} < t \leq t_p.$$

We also define a function $f^* : \mathcal{X} \to \mathcal{Y}$ by

$$f^*(x) = \begin{cases} y_p & \text{if } x \in B_p, \\ \bar{y} & \text{otherwise.} \end{cases}$$

This function is simple hence measurable. From now, we will suppose that the event $\mathcal{B}$ is met. For simplicity, we will denote by $\hat{Y}_t := f_t(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t)$ the prediction of the learning rule at time t. For any $p \geq 1$, because $\mathcal{E}_p \cap \mathcal{F}_p$ is met, for all $1 \leq u \leq N_p(T_p)$, we have $t_{p-1} < T_p(u) \leq T_p$, and $X_{T_p(u)} \in B_p$. Hence, by construction, we have $\hat{Y}_{T_q(u)} = y_q^u$ and we can write

$$\begin{aligned} \sum_{t=1}^{T_p} \ell(\hat{Y}_t, Y_t) &\geq \sum_{t=t_{p-1}+1}^{T_p} \ell(\hat{Y}_t, Y_t) \\ &\geq \sum_{u=1}^{N_p(T_p)} \ell(\hat{Y}_{T_p(u)}, Y_{T_p(u)}) \\ &= \sum_{u=1}^{\tau} \ell(f_{T_p(u)}\left(\mathbb{X}_{\leq T_p(u)-1}, \mathbb{Y}_{\leq T_p(u)-1}, X_{T_p(u)}\right), y_p^u). \end{aligned}$$

Now note that because the construction was similar to the construction of $g^p_\cdot$, we have $\mathbb{Y}_{\leq T_p(u)-1} = \left\{\mathbb{Y}_{\leq t_{p-1}}(\mathbb{X}), \mathbb{Y}_{t_{p-1}<t\leq T_p(u)-1}\left(\mathbb{X}, \{y_p^i\}_{i=1}^{u-1}\right)\right\}$, i.e., $\hat{Y}_{T_p(u)}$ coincides with the prediction $g_u^p(\{y_p^i\}_{i=1}^{u-1})$ provided that g_u^p precisely used the realization $\mathbb{X}$. Hence, conditioned on $\mathcal{B}$

for all $u \leq \tau_p$, $\hat{Y}_{T_p(u)}$ has the same distribution as $g_u^p(\boldsymbol{y}_{\boldsymbol{p}}^{\leq u-1})$. Therefore we obtain

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{T_p}\ell(\hat{Y}_t, Y_t) - \frac{1}{\tau}\sum_{u=1}^{\tau}\ell(y_p, y_p^u)\middle|\mathcal{B}\right] &\geq \mathbb{E}\left[\frac{1}{\tau}\sum_{u=1}^{\tau}\left(\ell(g_u^p(\hat{Y}_{T_p(u)}, y_p^u) - \ell(y_p, y_p^u)\right)\middle|\mathcal{B}\right] \\
&= \mathbb{E}\left[\frac{1}{\tau}\sum_{u=1}^{\tau}\left(\ell(g_u^p(\boldsymbol{y}_{\boldsymbol{p}}^{\leq u-1}), y_p^u) - \ell(y_p, y_p^u)\right)\right] \\
&\geq \eta.
\end{aligned}$$

We now turn to the loss obtained by the simple function f^*. By construction, assuming that the event $\mathcal{B}$ is met, we have

$$\sum_{t=1}^{T_p}\ell(f^*(X_t), Y_t) \leq \bar{\ell}t_{p-1} + \sum_{u=1}^{N_p(T_p)}\ell(f^*(X_{T_p(u)}), y_p^u) = \bar{\ell}t_{p-1} + \sum_{u=1}^{\tau}\ell(y_p, y_p^u).$$

Recalling that $T_p > \mu t_{p-1} \geq \frac{8\bar{\ell}}{\epsilon\eta}t_{p-1}$ and noting that $\tau = N_p(T_p) \geq \frac{\epsilon}{4}T_p$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sup_{t_{p-1}<T\leq t_p}\frac{1}{T}\sum_{t=1}^{T}(\ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t))\middle|\mathcal{B}\right] \\
&\geq \mathbb{E}\left[\frac{\tau}{T_p}\frac{1}{\tau}\left(\sum_{t=1}^{T}\ell(\hat{Y}_t, Y_t) - \sum_{u=1}^{\tau}\ell(y_p, y_p^u)\right) - \bar{\ell}\frac{t_{p-1}}{T_p}\middle|\mathcal{B}\right] \\
&\geq \frac{\epsilon}{4}\mathbb{E}\left[\frac{1}{\tau}\sum_{t=1}^{T_p}\ell(\hat{Y}_t, Y_t) - \frac{1}{\tau}\sum_{u=1}^{\tau}\ell(y_p, y_p^u)\middle|\mathcal{B}\right] - \frac{\epsilon\eta}{8} \\
&\geq \frac{\epsilon\eta}{8}.
\end{aligned}$$

Because this holds for any $p \geq 1$, Fatou lemma yields

$$\begin{aligned}
&\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t)\right] \\
&\geq \mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}(\ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t))\middle|\mathcal{B}\right]\mathbb{P}[\mathcal{B}] \\
&\geq \frac{\epsilon^2\eta}{32}.
\end{aligned}$$

Hence, we do note have almost surely $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 0$. This shows that $\mathbb{X} \notin \textsc{Solar}$, which in turn implies $\textsc{Solar} \subset \text{CS}$. This ends the proof that $\textsc{Solar} \subset \text{CS}$. The proof that $\text{CS} \subset \textsc{Solar}$ and the construction of an optimistically universal learning rule for adversarial regression is deferred to Section 4.7 where we give a stronger result which also holds for unbounded losses. Note that generalizing Theorem 4.8 to adversarial responses already shows that $\text{CS} \subset \textsc{Solar}$ and provides an optimistically universal learning rule when the loss ℓ is a metric.

4.11 Appendix C: Proofs of Section 4.6

4.11.1 Proof of Theorem 4.4

We first show that there exists $t_1 \geq 1$ such that for any $t \geq t_1$, with high probability, for all $i \in I_t$,

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \leq L_{t,i} + 3\ln^2 t\sqrt{t}.$$

For any $t \geq 0$, note that we have $\hat{\ell}_t = \mathbb{E}[\ell(\hat{Y}_t, Y_t) \mid \mathbb{Y}_{\leq t}]$. We define the instantaneous regret $r_{t,i} = \hat{\ell}_t - \ell(y^i, Y_t)$. We now define $w'_{t-1,i} := e^{\eta_{t-1}(\hat{L}_{t-1,i} - L_{t-1,i})}$ and pose $W_{t-1} = \sum_{i\in I_t} w_{t-1,i}$ and $W'_{t-1} = \sum_{i\in I_{t-1}} w'_{t-1,i}$, i.e., which induces the most regret. We also denote the index $k_t \in I_t$ such that $\hat{L}_{t,k_t} - L_{t,k_t} = \max_{i\in I_t} \hat{L}_{t,i} - L_{t,i}$. We first note that for any $i, j \in I_t$, we have $\ell(y^i, Y_t) - \ell(y^j, Y_t) \leq \ell(y^i, y^0) + \ell(y^0, y^j) \leq 2\ln t$. Therefore, we also have $|r_{t,i}| \leq 2\ln t$. Hence, we can apply Hoeffding's lemma to obtain

$$\frac{1}{\eta_t}\ln\frac{W'_t}{W_{t-1}} = \frac{1}{\eta_t}\ln\sum_{i\in I_t}\frac{w_{t-1,i}}{W_{t-1}}e^{\eta_t r_{t,i}} \leq \frac{1}{\eta_t}\left(\eta_t\sum_{i\in I_t} r_{t,i}\frac{w_{t-1,i}}{W_{t-1}} + \frac{\eta_t^2(4\ln t)^2}{8}\right) = 2\eta_t\ln^2 t.$$

The same computations as in the proof of Lemma 4.1 then show that

$$\begin{aligned}\frac{1}{\eta_t}\ln\frac{w_{t-1,k_{t-1}}}{W_{t-1}} - \frac{1}{\eta_{t+1}}\ln\frac{w_{t,k_t}}{W_t} \leq 2\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\ln(1+\ln(t+1)) + \frac{|I_{t+1}| - |I_t|}{\eta_t\sum_{i\in I_t} w_{t,i}} \\ + (\hat{L}_{t-1,k_{t-1}} - L_{t-1,k_{t-1}}) - (\hat{L}_{t,k_t} - L_{t,k_t}) + 2\eta_t\ln^2 t. \quad (4.4)\end{aligned}$$

First suppose that we have $\sum_{i\in I_t} w_{t,i} \leq 1$. Similarly to Lemma 4.1, we obtain $\hat{L}_{t,k_t} - L_{t,k_t} \leq 0$. Otherwise, let $t' = \min\{1 \leq s \leq t : \forall s \leq s' \leq t, \sum_{i\in I_{s'}} w_{s',i} \geq 1\}$. We sum Eq (4.4) for $s = t', \ldots, t$ which gives

$$\begin{aligned}\frac{1}{\eta_1}\ln\frac{w_{t'-1,k_{t'-1}}}{W_{t'-1}} - \frac{1}{\eta_{t+1}}\ln\frac{w_{t,k_t}}{W_t} \leq \frac{2}{\eta_{t+1}}\ln(1+\ln(t+1)) + \frac{|I_{t+1}|}{\eta_t} \\ + (\hat{L}_{t'-1,k_{t'-1}} - L_{t'-1,k_{t'-1}}) - (\hat{L}_{t,k_t} - L_{t,k_t}) + 2\sum_{s=t'}^{t}\eta_s\ln^2 s.\end{aligned}$$

Similarly as in Lemma 4.1, we have $\frac{w_{t,k_t}}{W_t} \leq 1$, $\frac{w_{t'-1,k_{t'-1}}}{W_{t'-1}} \geq \frac{1}{1+\ln t}$ and $\hat{L}_{t'-1,k_{t'-1}} - L_{t'-1,k_{t'-1}} \leq 0$. Finally, using the fact that $\sum_{s=1}^{t}\frac{1}{\sqrt{s}} \leq 2\sqrt{t}$, we obtain

$$\hat{L}_{t,k_t} - L_{t,k_t} \leq \ln(1+\ln(t+1))(4+8\sqrt{t+1}) + 4(1+\ln(t+1))\sqrt{t} + \ln^2 t\sqrt{t} \leq 2\ln^2 t\sqrt{t},$$

for all $t \geq t_0$ where t_0 is a fixed constant, and as a result, for all $t \geq t_0$ and $i \in I_t$, we have $\hat{L}_{t,i} - L_{t,i} \leq 2\ln^2 t\sqrt{t}$.

Now note that $|\ell(\hat{Y}_t, Y_t) - \mathbb{E}[\ell(\hat{Y}_t, Y_t) \mid \mathbb{Y}_{\leq t}]| \leq 2\ln t$ because for all $i \in I_t$, we have $\ell(y^i, y^0) \leq \ln t$. Hence, we can apply Hoeffding-Azuma inequality to the variables $\ell(\hat{Y}_t, Y_t) - \hat{\ell}_t$

that form a sequence of differences of a martingale, which yields

$$\mathbb{P}\left[\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) > \hat{L}_{t,i} + u\right] \leq e^{-\frac{u^2}{8t\ln^2 t}}.$$

Hence, for $t \geq t_0$ and $i \in I_t$, with probability $1-\delta$, we have

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \leq \hat{L}_{t,i} + \ln t\sqrt{2t\ln\frac{1}{\delta}} \leq L_{t,i} + 2\ln^2 t\sqrt{t} + \ln t\sqrt{2t\ln\frac{1}{\delta}}.$$

Therefore, since $|I_t| \leq 1 + \ln t$, by union bound with probability $1 - \frac{1}{t^2}$ we obtain that for all $i \in I_t$,

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \leq L_{t,i} + 2\ln^2 t\sqrt{t} + \ln t\sqrt{2t\ln(1+\ln t)} + \ln t\sqrt{4t\ln t} \leq 3\ln^2 t\sqrt{t}$$

for all $t \geq t_1$ where $t_1 \geq t_0$ is a fixed constant. Now because $\sum_{t\geq 1}\frac{1}{t^2} < \infty$, the Borel-Cantelli lemma implies that almost surely, there exists $\hat{t} \geq 0$ such that

$$\forall t \geq \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \leq L_{t,i} + 3\ln^2 t\sqrt{t}.$$

We denote by $\mathcal{A}$ this event. Now let $y \in \mathcal{Y}$, $\epsilon > 0$ and consider $i \geq 0$ such that $\ell(y^i, y) < \epsilon$. On the event $\mathcal{A}$, we have for all $t \geq \max(\hat{t}, t_i)$,

$$\sum_{s=t_i}^{t} \ell(\hat{Y}_s, Y_s) \leq \sum_{s=t_i}^{t} \ell(y^i, Y_s) + 3\ln^2 t\sqrt{t} \leq \sum_{s=t_i}^{t} \ell(y, Y_s) + \epsilon t + 3\ln^2 t\sqrt{t}.$$

Therefore, $\limsup_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t}\Big(\ell(\hat{Y}_s, Y_s) - \ell(y, Y_s)\Big) \leq \epsilon$ on $\mathcal{A}$. Because this holds for any $\epsilon > 0$ we finally obtain $\limsup_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t}\Big(\ell(\hat{Y}_s, Y_s) - \ell(y, Y_s)\Big) \leq 0$ on the event $\mathcal{A}$ of probability one, which holds for all $y \in \mathcal{Y}$ simultaneously. This ends the proof of the theorem.

4.11.2 Proof of Corollary 4.3

We denote by $g.$ the learning rule on values $\mathcal{Y}$ for mean estimation described in Theorem 4.4. Because processes in $\mathbb{X} \in \mathrm{FS}$ visit only finite number of different instance points in $\mathcal{X}$ almost surely, we can simply perform the learning rule $g.$ on each sub-process $\mathbb{Y}_{\{t:X_t=x\}}$ separately for any $x \in \mathcal{X}$. Note that the learning rule $g.$ does not explicitely re-use past randomness for its prediction. Hence, we will not specify that the randomness used for all learning rules—for each x visited by $\mathbb{X}$—should be independent. Let us formally describe our learning rule. Consider a sequence $\boldsymbol{x}_{\leq t-1}$ of instances in $\mathcal{X}$ and $\boldsymbol{y}_{\leq t-1}$ of values in $\mathcal{Y}$. We denote by $S_{t-1} = \{x : \boldsymbol{x}_{\leq t-1} \cap \{x\} \neq \emptyset\}$ the support of $\boldsymbol{x}_{\leq t-1}$. Further, for any $x \in S_{t-1}$, we denote

$N_{t-1}(x) = \sum_{u \leq t-1} \mathbb{1}_{x_u = x}$ the number of times that the specific instance x was visited by the sequence $\boldsymbol{x}_{\leq t-1}$. Last, for any $x \in S_{t-1}$, we denote $\boldsymbol{y}^x_{\leq N(x)}$ the values $\boldsymbol{y}_{\{u \leq t : X_u = x\}}$ obtained when the instance was precisely x in the sequence $\boldsymbol{x}_{\leq t-1}$, ordered by increasing time u. We are now ready to define our learning rule f_t at time t. Given a new instance point x_t, we pose

$$f_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{y}_{\leq t-1}, x_t) = \begin{cases} g_{N_{t-1}(x)+1}(\boldsymbol{y}^x_{\leq N_{t-1}(x)}) & \text{if } x_t \in S_{t-1}, \\ g_1(\emptyset) & \text{otherwise.} \end{cases}$$

Recall that for any $u \geq 1$, g_u uses some randomness. The only subtlety is that at each iteration $t \geq 1$ of the learning rule $f_\cdot$, the randomness used by the subroutine call to $g_\cdot$ should be independent from the past history. We now show that $f_\cdot$ is universally consistent for adversarial regression under all processes $\mathbb{X} \in \mathrm{FS}$.

Let $\mathbb{X} \in \mathrm{FS}$. For simplicity, we will denote by $\hat{Y}_t$ the prediction of the learning rule $f_\cdot$ at time t. We denote $S = \{x : \{x\} \cap \mathbb{X} \neq \emptyset\}$ the random support of $\mathbb{X}$. By hypothesis, we have $|S| < \infty$ with probability one. Denote by $\mathcal{E}$ this event. We now consider a specific realization $\boldsymbol{x}$ of $\mathbb{X}$ falling in the event $\mathcal{E}$. Then, S is a fixed set. We also denote $\tilde{S} := \{x \in S : \lim_{t \to \infty} N_t(x) = \infty\}$ the instances which are visited an infinite number of times by the sequence $\boldsymbol{x}$. Now, we can write for any function $f : \mathcal{X} \to \mathcal{Y}$,

$$\begin{aligned} \sum_{t=1}^{T} \Big(\ell(\hat{Y}_t, Y_t) - \ell(f(x_t), Y_t) \Big) &= \sum_{x \in S} \sum_{u=1}^{N_t(x)} \big(\ell(g_u(\mathbb{Y}^x_{\leq u-1}), Y^x_u) - \ell(f(x), Y_u) \big) \\ &\leq \sum_{s \in S \setminus \tilde{S}} \bar{\ell} |\{t \geq 1 : x_t = x\}| + \sum_{s \in \tilde{S}} \sum_{u=1}^{N_t(x)} \big(\ell(g_u(\mathbb{Y}^x_{\leq u-1}), Y^x_u) - \ell(f(x), Y_u) \big) . \end{aligned}$$

Now, because the randomness in $g_\cdot$ was taken independently from the past at each iteration, we can apply directly Theorem 4.4. For all $x \in \tilde{S}$, with probability one, for all $y^x \in \mathcal{Y}$,

$$\limsup_{t' \to \infty} \frac{1}{t'} \sum_{u=1}^{t'} \big(\ell(g_u(\mathbb{Y}^x_{\leq u-1}), Y^x_u) - \ell(y^x, Y_u) \big) \leq 0.$$

We denote by $\mathcal{E}_x$ this event. Then, on the event $\bigcap_{x \in \tilde{S}} \mathcal{E}_x$ of probability one, we have for any measurable function $f : \mathcal{X} \to \mathcal{Y}$,

$$\begin{aligned} &\limsup_{T \to \infty} \frac{1}{T} \Big(\ell(\hat{Y}_t, Y_t) - \ell(f(x_t), Y_t) \Big) \\ &\qquad \leq \sum_{s \in \tilde{S}} \limsup_{T \to \infty} \frac{1}{T} \sum_{u=1}^{N_t(x)} \big(\ell(g_u(\mathbb{Y}^x_{\leq u-1}), Y^x_u) - \ell(f(x), Y_u) \big) \\ &\qquad \leq \sum_{s \in \tilde{S}} \limsup_{T \to \infty} \frac{1}{N_t(x)} \sum_{u=1}^{N_t(x)} \big(\ell(g_u(\mathbb{Y}^x_{\leq u-1}), Y^x_u) - \ell(f(x), Y_u) \big) \leq 0. \end{aligned}$$

As a result, averaging on realisations of $\mathbb{X}$, we obtain that with probability one, we have that $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f_\cdot, f) \leq 0$ for all measurable functions $f : \mathcal{X} \to \mathcal{Y}$. Note that this is stronger than

the notion of universal consistency which we defined in Section 3.2, where we ask that for all measurable function $f : \mathcal{X} \to \mathcal{Y}$, we have almost surely $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f_\cdot, f) \leq 0$. In particular, this shows that FS $\subset$ SOLAR-U. As result SOLAR-U = FS and $f_\cdot$ is optimistically universal. This ends the proof of the result.

4.11.3 Proof of Theorem 4.12

We first show that mean-estimation is not achievable. To do so, let $f_\cdot$ be a learning rule. For simplicity, we will denote by $\hat{Y}_t$ its prediction at step t. We aim to construct a process $\mathbb{Y}$ on $\mathbb{R}$ and a value $y^* \in \mathbb{R}$ such that with non-zero probability we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y^*, Y_t) > 0.$$

We now pose $\beta := \frac{2\alpha}{\alpha-1} > 2$. For any sequence $\boldsymbol{b} := (b_t)_{t\geq 1}$ in $\{-1, 1\}$, we consider the following process $\mathbb{Y}^{\boldsymbol{b}}$ such that for any $t \geq 1$ we have $Y_t^{\boldsymbol{b}} = 2^{\beta t} b_t$. Let $\boldsymbol{B} := (B_t)_{t\geq 1}$ be an i.i.d. sequence of Rademacher random variables, i.e., such that $B_1 = 1$ (resp. $B_1 = -1$) with probability $\frac{1}{2}$. We consider the random variables $e_t := \mathbb{1}_{\hat{Y}_t \cdot Y_t \leq 0}$ which intuitively correspond to flags for large mistakes of the learning rule $f_\cdot$ at time t. Because $f_\cdot$ is an online learning rule, we have

$$\mathbb{E}[e_t \mid \mathbb{Y}_{\leq t-1}] = \mathbb{E}_{\hat{Y}_t}\left[\mathbb{E}_{B_t}[\mathbb{1}_{\hat{Y}_t \cdot Y_t \leq 0} \mid \hat{Y}_t]\right] = \mathbb{E}_{\hat{Y}_t}\left[\mathbb{1}_{\hat{Y}_t = 0} + \frac{1}{2}\mathbb{1}_{\hat{Y}_t \neq 0}\right] \geq \frac{1}{2}.$$

where the expectation $\mathbb{E}_{\hat{Y}_t}$ refers to the expectation on the randomness of the rule f_t. As a result, the random variables $e_t - \frac{1}{2}$ form a sequence of differences of a sub-martingale bounded by $\frac{1}{2}$ in absolute value. By the Azuma-Hoeffding inequality, we obtain $\mathbb{P}\left[\sum_{t=1}^{T} e_t \leq \frac{T}{4}\right] \leq e^{-T/8}$. Because $\sum_{t\geq 1} e^{-t/8} < \infty$, the Borel-Cantelli lemma implies that on an event $\mathcal{E}$ of probability one, we have $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} e_t \geq \frac{1}{4}$. As a result, there exists a specific realization $\boldsymbol{b}$ of $\boldsymbol{B}$ such that on an event $\tilde{\mathcal{E}}$ of probability one, we have $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} e_t \geq \frac{1}{4}$. Note that the sequence $\mathbb{Y}^{\boldsymbol{b}}$ is now deterministic. Then, writing $e_t = e_t\mathbb{1}_{Y_t>0} + e_t\mathbb{1}_{Y_t<0}$, we obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} e_t\mathbb{1}_{Y_t>0} + \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} e_t\mathbb{1}_{Y_t<0} \geq \frac{1}{4}.$$

Without loss of generality, we can suppose that $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\hat{Y}_t \cdot Y_t \leq 0}\mathbb{1}_{Y_t>0} \geq \frac{1}{8}$. We now pose $y^* = 1$. In the other case, we pose $y^* = -1$. We now compute for any $T \geq 1$ such that $\hat{Y}_t \cdot Y_t \leq 0$ and $Y_t > 0$,

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} (\ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y^*, Y_t)) &\geq \frac{\ell(0, 2^{\beta^T}) - \ell(1, 2^{\beta^T})}{T} - \frac{1}{T}\sum_{t=1}^{T-1} \ell(1, -2^{\beta^t}). \\
&= \frac{\alpha}{T} 2^{(\alpha-1)\beta^T} + O\left(\frac{1}{T} 2^{(\alpha-2)\beta^T}\right) - 2^{\alpha(1+\beta^{T-1})} \\
&= \frac{\alpha}{T} 2^{2\alpha\beta^{T-1}}(1 + o(1)).
\end{aligned}$$

Because, by construction $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\hat{Y}_t \cdot Y_t \leq 0}\mathbb{1}_{Y_t>0} \geq \frac{1}{8}$, we obtain

$$\limsup \frac{1}{T}\sum_{t=1}^{T} (\ell(f_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y^*, Y_t)) = \infty,$$

on the event $\tilde{E}$ of probability one. This end the proof that mean-estimation is not achievable. Because mean-estimation is the easiest regression setting, this directly implies SOLAR-U $= \emptyset$. Formally, let $\mathbb{X}$ a process on $\mathcal{X}$. and $f.$ a learning rule for regression. We consider the same processes $\mathbb{Y}^{\boldsymbol{B}}$ where $\boldsymbol{B}$ is i.i.d. Rademacher and independent from $\mathbb{X}$. The same proof shows that there exists a realization $\boldsymbol{b}$ for which we have almost surely $\mathcal{L}_{(\mathbb{X},\mathbb{Y})}(f., f^* := y^*) = \infty$, where $f^* = y^*$ denotes the constant function equal to y^* where $y^* \in \mathbb{R}$ is the value constructed as above. Hence, $\mathbb{X} \notin$ SOLAR-U, and as a result, SOLAR-U $= \emptyset$.

4.11.4 Proof of Proposition 4.1

Suppose that there exists an online learning rule $g.$ for mean-estimation. In the proof of Corollary 4.3, instead of using the learning rule for mean-estimation for metric losses introduced in Theorem 4.4, we can use the learning rule $g.$ to construct the learning rule $f.$ for adversarial regression on FS instance processes, which simply performs $f.$ separately on each subprocess $\mathbb{Y}_{t:X_t=x}$ with the same instance $x \in \mathcal{X}$ for all visited $x \in \mathcal{X}$ in the process $\mathbb{X}$. The same proof shows that because almost surely $\mathbb{X}$ visits a finite number of different instances, $f.$ is universally consistent under any process $\mathbb{X} \in$ FS. Hence, FS $\subset$ SOLAR-U. Because SOLAR-U $\subset$ SOUL $=$ FS, we obtain directly SOLAR-U $=$ FS and $f.$ is optimistically universal.

On the other hand, if mean-estimation with adversarial responses is not achievable, we can use similar arguments as for the proof of Theorem 4.12. Let $f.$ a learning rule for regression, and consider the following learning rule $g.$ for mean-estimation. We first draw a process $\tilde{\mathbb{X}}$ with same distribution as $\mathbb{X}$. Then, we pose

$$g_t(\boldsymbol{y}_{\leq t-1}) := f_t(\tilde{\mathbb{X}}_{\leq t-1}, \boldsymbol{y}_{\leq t-1}, \tilde{X}_t).$$

Then, because mean-estimation is not achievable, there exists an adversarial process $\mathbb{Y}$ on $(\mathcal{Y}, \ell)$ such that with non-zero probability,

$$\limsup \frac{1}{T}\sum_{t=1}^{T} (\ell(g_t(\mathbb{Y}_{\leq t-1}), Y_t) - \ell(y^*, Y_t)) > 0.$$

Then, we obtain that with non-zero probability, $\mathcal{L}_{(\tilde{\mathbb{X}},\mathbb{Y})} > 0$. Hence, $f.$ is not universally consistent. Note that the "bad" process $\mathbb{Y}$ is not correlated with $\tilde{\mathbb{X}}$ in this construction.

4.12 Appendix D: Proofs of Section 4.7

4.12.1 Proof of Theorem 4.13

Let $(x^k)_{k\geq 0}$ a sequence of distinct points of $\mathcal{X}$. Now fix a value $y_0 \in \mathcal{Y}$ and construct a sequence of values y_k^1, y_k^2 for $k \geq 1$ such that $\ell(y_k^1, y_k^2) \geq c_\ell 2^{k+1}$. Because $\ell(y_k^1, y_k^2) \leq$

$c_\ell \ell(y_0, y_k^1) + c_\ell \ell(y_0, y_k^2)$, there exists $i_k \in \{1, 2\}$ such that $\ell(y_0, y_k^{i_k}) \geq 2^k$. For simplicity, we will now write $y_k := y_k^{i_k}$ for all $k \geq 1$. We define

$$t_k = \left\lfloor \sum_{l=1}^{k} \ell(y_0, y_l) \right\rfloor.$$

This forms an increasing sequence of times because $t_{k+1} - t_k \geq \ell(y_0, y_{k+1}) \geq 1$. Consider the deterministic process $\mathbb{X}$ that visits x^k at time t_k and x^0 otherwise, i.e., such that

$$X_t = \begin{cases} x^k & \text{if } t = t_k, \\ x^0 & \text{otherwise.} \end{cases}$$

The process $\mathbb{X}$ visits $\mathcal{X} \setminus \{x^0\}$ a sublinear number of times. Hence we have for any measurable set A:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_A(X_t) = \begin{cases} 1 & \text{if } x^0 \in A \\ 0 & \text{otherwise.} \end{cases}$$

As a result, $\mathbb{X} \in \text{CRF}$. We will now show that universal learning under $\mathbb{X}$ with the first moment condition on the responses is not achievable. For any sequence $b := (b_k)_{k \geq 1}$ of binary variables $b_k \in \{0, 1\}$, we define the function $f_b^* : \mathcal{X} \to \mathcal{Y}$ such that

$$f_b^*(x^k) = \begin{cases} y_0 & \text{if } b_k = 0, \\ y_k & \text{otherwise,} \end{cases} \quad k \geq 0 \quad \text{and} \quad f_b^*(x) = y_0 \text{ if } x \notin \{x_k, k \geq 0\}.$$

These functions are simple, hence measurable. We will first show that for any binary sequence b, the function f_b^* satisfies the moment condition on the target functions. Indeed, we note that for any $T \geq t_1$, with $k := \max\{l \geq 1 : t_l \leq T\}$, we have

$$\frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f_b^*(X_t)) \leq \frac{1}{T} \sum_{l=1}^{k} \ell(y_0, y_k) \leq \frac{t_k + 1}{T} \leq \frac{T + 1}{T}.$$

Therefore, $\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f_b^*(X_t)) \leq 1$. We now consider any online learning rule $f_\cdot$. Let $B = (B_k)_{k \geq 1}$ be an i.i.d. sequence of Bernoulli variables independent from the learning rule randomness. For any $k \geq 1$, denoting by $\hat{Y}_{t_k} := f_{t_k}(\mathbb{X}_{\leq t_k - 1}, f_B^*(\mathbb{X}_{\leq t_k - 1}), X_{t_k})$ we have

$$\mathbb{E}_{B_k} \ell(\hat{Y}_{t_k}, f_B^*(X_{t_k})) = \frac{\ell(\hat{Y}_{t_k}, y_0) + \ell(\hat{Y}_{t_k}, y_k)}{2} \geq \frac{1}{2c_\ell} \ell(y_0, y_k).$$

In particular, taking the expectation over both B and the learning rule, we obtain

$$\mathbb{E}\left[\frac{1}{t_k} \sum_{t=1}^{t_k} \ell(f_t(\mathbb{X}_{\leq t-1}, f_B^*(\mathbb{X}_{\leq t-1}), X_t), f_B^*(X_t)) \right] \geq \frac{1}{2c_\ell t_k} \sum_{l=1}^{k} \ell(y_0, y_k) \geq \frac{1}{2c_\ell}.$$

As a result, using Fatou's lemma we obtain

$$\begin{aligned}
&\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(f_t(\mathbb{X}_{\leq t-1},f_B^*(\mathbb{X}_{\leq t-1}),X_t),f_B^*(X_t))\right]\\
&\geq \limsup_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f_t(\mathbb{X}_{\leq t-1},f_B^*(\mathbb{X}_{\leq t-1}),X_t),f_B^*(X_t))\right]\\
&\geq \frac{1}{2c_\ell}.
\end{aligned}$$

Therefore, the learning rule $f_.$ is not consistent under $\mathbb{X}$ for all target functions of the form f_b^* for some sequence of binary variables b. Indeed, otherwise for all binary sequence $b=(b_k)_{k\geq 1}$, we would have $\mathbb{E}_{\mathbb{X}}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(f_t(\mathbb{X}_{\leq t-1},f_b^*(\mathbb{X}_{\leq t-1}),X_t),f_b^*(X_t))\right]=0$ and as a result

$$\mathbb{E}_B\mathbb{E}_{\mathbb{X}}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(f_t(\mathbb{X}_{\leq t-1},f_B^*(\mathbb{X}_{\leq t-1}),X_t),f_B^*(X_t))\right]=0.$$

This ends the proof of the theorem.

4.12.2 Proof of Lemma 4.5

It suffices to prove that empirical integrability implies the latter property. We pose $\epsilon_i=2^{-i}$ for any $i\geq 0$. By definition, there exists an event $\mathcal{E}_i$ of probability one such that on $\mathcal{E}_i$ we have

$$\exists M_i\geq 0,\quad \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(y_0,Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M_i}\leq\epsilon_i.$$

As a result, on $\bigcap_{i\geq 0}\mathcal{E}_i$ of probability one, we obtain

$$\forall\epsilon>0,\exists M:=M_{\lceil\log_2\frac{1}{\epsilon}\rceil}\geq 0,\quad \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(y_0,Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M}\leq\epsilon.$$

This ends the proof of the lemma.

4.12.3 Proof of Theorem 4.1

Let $\mathbb{X}\in$ Soul and $f^*:\mathcal{X}\to\mathcal{Y}$ such that $f^*(\mathbb{X})$ is empirically integrable. By Lemma 4.5, there exists some value $y_0\in\mathcal{Y}$ such that on an event $\mathcal{A}$ of probability one, for all $\epsilon>0$ there exists $M_\epsilon\geq 0$ such that

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(y_0,f^*(X_t))\mathbb{1}_{\ell(y_0,f^*(X_t))\geq M_\epsilon}\leq\epsilon.$$

For any $M \geq 1$ we define the function f^*_M by

$$f^*_M(x) = \begin{cases} f^*(x) & \text{if } \ell(y_0, f^*(x)) \leq M, \\ y_0 & \text{otherwise.} \end{cases}$$

We know that 2C1NN is optimistically universal in the noiseless setting for bounded losses. Therefore, restricting the study to the output space $(B_\ell(y_0, M), \ell)$ we obtain that 2C1NN is consistent for f^*_M under $\mathbb{X}$, i.e.

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(2C1NN_t(\mathbb{X}_{t-1}, f^*_M(\mathbb{X}_{\leq t-1}), X_t), f^*_M(X_t)) = 0 \quad (a.s.).$$

For any $t \geq 1$, we denote $\phi(t)$ the representative used by the 2C1NN learning rule. We denote $\mathcal{E}_M$ the above event such that $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(f^*_M(X_{\phi(t)}), f^*_M(X_t)) = 0$. We now write for any $T \geq 1$ and $M \geq 1$,

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*(X_t)) \leq \frac{c_\ell^2}{T} \sum_{t=1}^{T} \ell(f^*_M(X_{\phi(t)}), f^*_M(X_t)) + \frac{c_\ell^2}{T} \sum_{t=1}^{T} \ell(f^*(X_t), f^*_M(X_t)) \\ + \frac{c_\ell}{T} \sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*_M(X_{\phi(t)})). \end{aligned}$$

We now note that by construction of the 2C1NN learning rule,

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*_M(X_{\phi(t)})) &= \frac{1}{T} \sum_{u=1}^{T} \ell(f^*(X_u), f^*_M(X_u)) |\{u < t \leq T : \phi(t) = u\}| \\ &\leq \frac{2}{T} \sum_{t=1}^{T} \ell(f^*(X_t), f^*_M(X_t)). \end{aligned}$$

Hence, we obtain

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*(X_t)) \leq \frac{c_\ell^2}{T} \sum_{t=1}^{T} \ell(f^*_M(X_{\phi(t)}), f^*_M(X_t)) \\ + \frac{c_\ell(2 + c_\ell)}{T} \sum_{t=1}^{T} \ell(y_0, f^*(X_t)) \mathbb{1}_{\ell(y_0, f^*(X_t)) > M}. \end{aligned}$$

As a result, on the event $\mathcal{A} \cap \bigcap_{M\geq 1} \mathcal{E}_M$ of probability one, for any $M \geq 1$, we obtain

$$\begin{aligned} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*(X_t)) \\ \leq c_\ell(2 + c_\ell) \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(y_0, f^*(X_t)) \mathbb{1}_{\ell(y_0, f^*(X_t)) \geq M}. \end{aligned}$$

In particular, if $\epsilon > 0$ we can apply this result with $M := \lceil M_\epsilon \rceil$, which shows that $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(f^*(X_{\phi(t)}), f^*(X_t)) \leq c_\ell(2 + c_\ell)\epsilon$. Because this holds for any $\epsilon > 0$ we finally obtain that on the event $\mathcal{A} \cap \bigcap_{M\geq 1} \mathcal{E}_M$ we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(f^*(X_\phi(t)), f^*(X_t)) = 0.$$

This ends the proof of the theorem.

4.12.4 Proof of Theorem 4.3

We first define the learning rule. Using Lemma 23 of [Han21a], let $\mathcal{T} \subset \mathcal{B}$ a countable set such that for all $\mathbb{X} \in \mathrm{CS}, A \subset \mathcal{B}$ we have

$$\inf_{G\in\mathcal{T}} \mathbb{E}[\hat{\mu}_{\mathbb{X}}(G \triangle A)] = 0.$$

Now let $(y^i)_{i\geq 0}$ be a dense sequence in $\mathcal{Y}$. For any $k \geq 0$, any indices $l_1, \ldots, l_k \in \mathbb{N}$ and any sets $A_1, \ldots, A_k \in \mathcal{T}$, we define the function $f_{\{l_1,\ldots,l_k\},\{A_1,\ldots,A_k\}} : \mathcal{X} \to \mathcal{Y}$ as

$$f_{\{l_1,\ldots,l_k\},\{A_1,\ldots,A_k\}}(x) = y^{\max\{0\leq j\leq k:\, x\in A_j\}}$$

where $A_0 = \mathcal{X}$. These functions are simple hence measurable. Because the set of such functions is countable, we enumerate these functions as $f^0, f^1 \ldots$ Without loss of generality, we suppose that $f^0 = y^0$. For any $i \geq 0$, we denote $k^i \geq 0$, $\{l_1^i, \ldots, l_{k^i}^i\}$ and $\{A_1^i, \ldots, A_{k^i}^i\}$ such that f^i was defined as $f^i := f_{\{l_1^i,\ldots,l_k^i\},\{A_1^i,\ldots,A_k^i\}}$. We now define a sequence of sets $(I_t)_{t\geq 1}$ of indices and a sequence of sets $(\mathcal{F}_t)_{t\geq 1}$ of measurable functions by $s^\epsilon := 1/(2 + c_1^\ell)$,

$$I_t := \{i \leq \ln t : \ell(y^{l_p^i}, y^0) \leq s^\epsilon \ln t,\ \forall 1 \leq p \leq k^i\} \quad \text{and} \quad \mathcal{F}_t := \{f^i : i \in I_t\}.$$

Then, clearly I_t is finite and $\bigcup_{t\geq 1} I_t = \mathbb{N}$. For any $i \geq 0$, we define $t_i = \min\{t : i \in I_t\}$. We are now ready to construct our learning rule. Let $\eta_t = \frac{1}{\ln t\sqrt{t}}$. Fix any sequences $(x_t)_{t\geq 1}$ in $\mathcal{X}$ and $(y_t)_{t\geq 1}$ in $\mathcal{Y}$. At step $t \geq 1$, after observing the values x_i for $1 \leq i \leq t$ and y_i for $1 \leq i \leq t-1$, we define for any $i \in I_t$ the loss $L_{t-1,i} := \sum_{s=t_i}^{t-1} \ell(f^i(x_s), y_s)$. For any $M \geq 1$ we define the function $\phi_M : \mathcal{Y} \to \mathcal{Y}$ such that

$$\phi_M(y) = \begin{cases} y & \text{if } \ell(y, y^0) < M, \\ y^0 & \text{otherwise.} \end{cases}$$

We now construct construct some weights $w_{t,i}$ for $t \geq 1$ and $i \in I_t$ recursively in the following way. Note that $I_1 = \{0\}$. Therefore, we pose $w_{0,0} = 1$. Now let $t \geq 2$ and suppose that $w_{s-1,i}$ have been constructed for all $1 \leq s \leq t-1$. We define

$$\hat{\ell}_s := \frac{\sum_{j\in I_s} w_{s-1,j}\ell(f^j(x_s), \phi_{s^\epsilon \ln s}(y_s))}{\sum_{j\in I_s} w_{s-1,j}}$$

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T
Output: Predictions $\hat{Y}_t$ for $t \leq T$
Construct the sequence of measurable functions $\{f^i, i \geq 0\}$ with $f^i = f_{\{l_1^i,\dots,l_k^i\},\{A_1^i,\dots,A_k^i\}}$
$I_t := \{i \leq \ln t, \ell(y^{l_p^i}, y^0) \leq s^\epsilon \ln t, \forall 1 \leq p \leq k^i\}, \mathcal{F}_t := \{f^i, i \in I_t\}, \eta_t := \frac{1}{\ln t \sqrt{t}}, t \geq 1$
$t_i = \min\{t : i \in I_t\}, i \geq 0$
$w_{0,0} := 1, \quad \hat{Y}_1 = y^0 (= f^0(X_0))$ `// Initialisation`
for $t = 2, \dots, T$ **do**
  $L_{t-1,i} = \sum_{s=t_i}^{t-1} \ell(f^i(X_s), \phi_{s^\epsilon \ln t}(Y_s)), \quad \hat{L}_{t-1,i} = \sum_{s=t_i}^{t-1} \hat{\ell}_s, \quad i \in I_t$
  $w_{t-1,i} := \exp(\eta_t(\hat{L}_{t-1,i} - L_{t-1,i})), \quad i \in I_t$
  $p_t(i) = \frac{w_{t-1,i}}{\sum_{j \in I_t} w_{t-1,j}}, \quad i \in I_t$
  $\hat{i}_t \sim p_t(\cdot)$ `// Function selection`
  $\hat{Y}_t = f^{\hat{i}_t}(X_t)$
  $\hat{\ell}_t := \frac{\sum_{j \in I_t} w_{t-1,j} \ell(f^j(X_s), \phi_{s^\epsilon \ln t}(Y_t)}{\sum_{j \in I_t} w_{t-1,j}}$
end

Algorithm 4.6: A learning rule for adversarial empirically integrable responses under CS processes.

and for any $i \in I_t$ we note $\hat{L}_{t-1,i} := \sum_{s=t_i}^{t-1} \hat{\ell}_s$. In particular, if $t_i = t$ we have $\hat{L}_{t-1,i} = L_{t-1,i} = 0$. The weights at time t are constructed as $w_{t-1,i} := e^{\eta_t(\hat{L}_{t-1,i} - L_{t-1,i})}$ for any $i \in I_t$. Last, let $\{\hat{i}_t\}_{t \geq 1}$ a sequence of independent random $\mathbb{N}-$valued variables such that

$$\mathbb{P}(\hat{i}_t = i) = \frac{w_{t-1,i}}{\sum_{j \in I_t} w_{t-1,j}}, \quad i \in I_t.$$

Finally, the prediction is defined as $\hat{y}_t := f^{\hat{i}_t}(x_t)$. The learning rule is summarized in Algorithm 4.6.

For simplicity, we will refer to the predictions of the learning rule as $(\hat{Y}_t)_{t \geq 1}$. Now consider a process $(\mathbb{X}, \mathbb{Y})$ with $\mathbb{X} \in \mathrm{CS}$ and such that $\mathbb{Y}$ is empirically integrable. By Lemma 4.5, there exists $y_0 \in \mathcal{Y}$ such that on an event $\mathcal{A}$ of probability one, for any $\epsilon > 0$, there exists $M_\epsilon \geq 0$ with $\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \ell(y_0, Y_t) \mathbb{1}_{\ell(y_0, Y_t) \geq M_\epsilon} \leq \epsilon$. We will now denote $\tilde{\mathbb{Y}}$ the process defined by $\tilde{Y}_t = \phi_{s^\epsilon \ln t}(Y_t)$ for all $t \geq 1$. Then, for any $i \in I_t$, note that using the generalized-metric loss identity we have

$$0 \leq \ell(f^i(x_t), \tilde{Y}_t) \leq 2\ell(f^i(x_t), y^0) + c_1^\ell \ell(y^0, \tilde{Y}_t) \leq 2 \ln t,$$

by construction of the set I_t. As a result, for any $i, j \in I_t$, we obtain $|\ell(f^i(x_t), \tilde{Y}_t^M) - \ell(f^j(x_t) - \tilde{Y}_t^M)| \leq 2 \ln t$. Hence, we can use the same proof as for Theorem 4.4 and show that almost surely, there exists $\hat{t} \geq 1$ such that

$$\forall t \geq \hat{t}, \forall i \in I_t, \quad \sum_{s=t_i}^t \ell(\hat{Y}_s, \tilde{Y}_s^M) \leq L_{t,i} + 3 \ln^2 t \sqrt{t}.$$

We denote by $\mathcal{B}$ this event. Now let $f : \mathcal{X} \to \mathcal{Y}$ to which we compare the predictions of our learning rule. For any $M \geq 1$, the function $\phi_M \circ f$ is measurable and has values in the

ball $B_\ell(y_0, M)$ where the loss is bounded by $(2 + c_1^\ell)M$. Hence, by Lemma 24 from [Han21a] because $\mathbb{X} \in \mathcal{C}_1$ we have

$$\inf_{i \geq 0} \mathbb{E}\left[\hat{\mu}_{\mathbb{X}}(\ell(\phi_M \circ f(\cdot), f^i(\cdot)))\right] = 0.$$

Now for any $k \geq 0$, let $i_k \geq 0$ such that $\mathbb{E}\left[\hat{\mu}_{\mathbb{X}}(\ell(\phi_M \circ f(\cdot), f^{i_k}(\cdot)))\right] < 2^{-2k}$. By Markov inequality, we have

$$\mathbb{P}\left[\hat{\mu}_{\mathbb{X}}(\ell(\phi_M \circ f(\cdot), f^{i_k}(\cdot))) < 2^{-k}\right] \geq 1 - 2^{-k}.$$

Because $\sum_k 2^{-k} < \infty$, the Borel-Cantelli lemma implies that almost surely there exists $\hat{k}$ such that for any $k \geq \hat{k}$, the above inequality is met. We denote $\mathcal{E}_M$ this event. On the event $\mathcal{B} \cap \mathcal{E}_M$ of probability one, for $k \geq \hat{k}$ and any $T \geq \max(t_{i_k}, \hat{t})$ we have for any $\epsilon > 0$,

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)\Big) \\
&= \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, \tilde{Y}_t) - \ell(f^{i_k}(X_t), \tilde{Y}_t) + \frac{1}{T}\sum_{t=1}^{T} \ell(f^{i_k}(X_t), \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t) \\
&\leq \frac{1}{T}\sum_{t=1}^{t_{i_k}-1} \ell(\hat{Y}_t, \tilde{Y}_t) + \frac{1}{T}\left(\sum_{t=t_{i_k}}^{T} \ell(\hat{Y}_t, \tilde{Y}_t) - L_{T,i_k}\right) + \frac{\epsilon}{T}\sum_{t=1}^{T} \ell(\phi_M \circ f(X_t), \tilde{Y}_t) \\
&\qquad + \frac{c_\epsilon^\ell}{T}\sum_{t=1}^{T} \ell(f^{i_k}(X_t), \phi_M \circ f(X_t)) \\
&\leq \frac{2\ln t_{i_k}}{T} + \frac{3\ln^2 T}{\sqrt{T}} + \epsilon c_1^\ell M + \frac{2\epsilon}{T}\sum_{t=1}^{T} \ell(y^0, \tilde{Y}_t) + \frac{c_\epsilon^\ell}{T}\sum_{t=1}^{T} \ell(f^{i_k}(X_t), \phi_M \circ f(X_t)) \\
&\leq \frac{2\ln t_{i_k}}{T} + \frac{3\ln^2 T}{\sqrt{T}} + \epsilon c_1^\ell M + \frac{2\epsilon}{T}\sum_{t=1}^{T} \ell(y^0, Y_t) + \frac{c_\epsilon^\ell}{T}\sum_{t=1}^{T} \ell(f^{i_k}(X_t), \phi_M \circ f(X_t)),
\end{aligned}$$

where in the last inequality we used the inequality $\ell(y^0, \tilde{Y}_t) \leq \ell(y^0, Y_t)$ by construction of $\tilde{Y}_t = \phi_{s^\epsilon \ln t}(Y_t)$. Now on the event $\mathcal{A}$, we have

$$\begin{aligned}
Z_1 := \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y^0, Y_t) &\leq c_1^\ell \ell(y_0, y^0) + 2\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y_0, Y_t) \\
&\leq c_1^\ell \ell(y_0, y^0) + 2M_1 + 2\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M_1} \\
&\leq c_1^\ell \ell(y_0, y^0) + 2(M_1 + 1) := \bar{Z}_1 < \infty.
\end{aligned}$$

Thus, on the event $\mathcal{A} \cap \mathcal{B} \cap \mathcal{E}_M$, for any $k \geq \hat{k}$ we have for any $\epsilon > 0$,

$$\limsup_{T} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)) \leq \epsilon c_1^\ell M + 2\epsilon \bar{Z}_1 + \frac{c_\epsilon^\ell}{2^k}.$$

Let $\delta > 0$. Now taking $\epsilon = \frac{\delta}{c_1^\ell M + 2\bar{Z}_1}$, we obtain that on the event $\mathcal{A} \cap \mathcal{B} \cap \mathcal{E}_M$, for any $k \geq \hat{k}$, we have $\limsup_T \frac{1}{T}\sum_{t=1}^T \ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)) \leq \delta + \frac{c_\epsilon^\ell}{2^k}$. This yields $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)) \leq \delta$. Because this holds for any $\delta > 0$ we obtain $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)) \leq 0$. Finally, on the event $\mathcal{A} \cap \mathcal{B} \cap \bigcap_{M=1}^\infty \mathcal{E}_M$ of probability one, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \Big(\ell(\hat{Y}_t, \tilde{Y}_t) - \ell(\phi_M \circ f(X_t), \tilde{Y}_t)\Big) \leq 0, \quad \forall M \geq 1,$$

where M is an integer. We now observe that on the event $\mathcal{A}$, the same guarantee for y_0 also holds for y^0. Indeed, let ϵ. For $\tilde{M}_\epsilon := 2(M_{\epsilon/3} + c_1^\ell \ell(y^0, y_0)) + c_1^\ell \ell(y_0, y^0)$ we have

$$\begin{aligned}
&\frac{1}{T}\sum_{T=1}^T \ell(y^0, Y_t)\mathbb{1}_{\ell(y^0,Y_t)\geq \tilde{M}_\epsilon} \\
&\leq c_1^\ell \ell(y^0, y_0)\frac{1}{T}\sum_{t=1}^T \mathbb{1}_{\ell(y^0,Y_t)\geq \tilde{M}_\epsilon} + \frac{2}{T}\sum_{T=1}^T \ell(y_0, Y_t)\mathbb{1}_{\ell(y^0,Y_t)\geq \tilde{M}_\epsilon} \\
&\leq c_1^\ell \ell(y^0, y_0)\frac{1}{T}\sum_{t=1}^T \mathbb{1}_{\ell(y_0,Y_t)\geq (\tilde{M}_\epsilon - c_1^\ell \ell(y_0,y^0))/2} + \frac{2}{T}\sum_{T=1}^T \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq (\tilde{M}_\epsilon - c_1^\ell \ell(y_0,y^0))/2} \\
&\leq \frac{3}{T}\sum_{t=1}^T \ell(y_0, Y_t)\mathbb{1}_{\ell(y_0,Y_t)\geq M_{\epsilon/3}}
\end{aligned}$$

Hence, we obtain $\limsup_{T\to\infty} \frac{1}{T}\sum_{T=1}^T \ell(y^0, Y_t)\mathbb{1}_{\ell(y^0,Y_t)\geq \tilde{M}_\epsilon} \leq \epsilon$. We now write

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^T \ell(\phi_M \circ f(X_t), \tilde{Y}_t) - \ell(f(X_t), Y_t) \\
&\quad\leq \frac{1}{T}\sum_{t=1}^T \big(\ell(y^0, Y_t) - \ell(f(X_t), Y_t)\big)\,\mathbb{1}_{\ell(f(X_t),y^0)\geq M}\mathbb{1}_{\ell(Y_t,y^0)\leq \ln t} \\
&\qquad\quad + \frac{1}{T}\sum_{t=1}^T \big(\ell(f(X_t), y^0) - \ell(f(X_t), Y_t)\big)\,\mathbb{1}_{\ell(f(X_t),y^0)\leq M}\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t} \\
&\quad\leq \frac{1}{T}\sum_{t=1}^T \big((1 + c_1^\ell/2)\ell(y^0, Y_t) - \ell(f(X_t), y^0)/2\big)\,\mathbb{1}_{\ell(f(X_t),y^0)\geq M} \\
&\qquad\quad + \frac{1}{T}\sum_{t=1}^T \big((1 + c_1^\ell/2)\ell(f(X_t), y^0) - \ell(y^0, Y_t)/2\big)\,\mathbb{1}_{\ell(f(X_t),y^0)\leq M}\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t} \\
&\quad\leq \frac{1 + c_1^\ell/2}{T}\sum_{t=1}^T \ell(y^0, Y_t)\mathbb{1}_{\ell(Y_t,y^0)\geq M/(2+c_1^\ell)} + \frac{(1 + c_1^\ell/2)M\exp\big\{2(1 + c_1^\ell/2)/s^\epsilon M\big\}}{T}.
\end{aligned}$$

As a result, on the event $\mathcal{A} \cap \mathcal{B} \cap \bigcap_{M=1}^\infty \mathcal{E}_M$, for any $M \geq 1$,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \ell(\phi_M \circ f(X_t), \tilde{Y}_t) - \ell(f(X_t), Y_t) \leq \limsup_{T\to\infty} \frac{1 + c_1^\ell/2}{T}\sum_{t=1}^T \ell(y^0, Y_t)\mathbb{1}_{\ell(Y_t,y^0)\geq M/(2+c_1^\ell)}.$$

Last, we compute

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(\hat{Y}_t,\tilde{Y}_t) &= \frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat{Y}_t,Y_t)-\ell(\hat{Y}_t,y^0)\Big)\,\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t}\\ &\leq \frac{1}{T}\sum_{t=1}^{T}\Big(2\ell(\hat{Y}_t,y^0)+c_1^\ell\ell(Y_t,y^0)\Big)\,\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t}\\ &\leq \frac{1}{T}\sum_{t=1}^{T}\big(\ln t+c_1^\ell\ell(Y_t,y^0)\big)\,\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t}\\ &\leq \frac{c_1^\ell+1/s^\epsilon}{T}\sum_{t=1}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t}.\end{aligned}$$

Note that for any $\epsilon>0$, we have on the event $\mathcal{A}$ that for any $M\geq 1$,

$$\begin{aligned}\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon \ln t} &\leq \limsup_{T\to\infty}\frac{1}{T}\sum_{t\geq e^{M/s^\epsilon}}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq M}\\ &= \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq M}.\end{aligned}$$

Hence, because this holds for any $M\geq 1$, if $\epsilon>0$ we can apply this to the integer $M:=\lceil\tilde{M}_\epsilon\rceil$ which yields $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon\ln t}\leq\epsilon$. This holds for any $\epsilon>0$. Hence we obtain on the event $\mathcal{A}$ that $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,y^0)\mathbb{1}_{\ell(Y_t,y^0)\geq s^\epsilon\ln t}\leq 0$, which implies that $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(\hat{Y}_t,\tilde{Y}_t)\leq 0$. Putting everything together, we obtain on $\mathcal{A}\cap\mathcal{B}\cap\bigcap_{M=1}^{\infty}\mathcal{E}_M$ that for any $M\geq 1$,

$$\begin{aligned}\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(f(X_t),Y_t) &\leq \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(\hat{Y}_t,\tilde{Y}_t)\\ &+\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,\tilde{Y}_t)-\ell(\phi_M\circ f(X_t),\tilde{Y}_t)\\ &+\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\phi_M\circ f(X_t),\tilde{Y}_t)-\ell(f(X_t),Y_t)\\ &\leq \limsup_{T\to\infty}\frac{1+c_1^\ell/2}{T}\sum_{t=1}^{T}\ell(y^0,Y_t)\mathbb{1}_{\ell(Y_t,y^0)\geq M(2+c_1^\ell)}.\end{aligned}$$

Because this holds for all $M\geq 1$, we can again apply this result to $M:=\lceil\tilde{M}_\epsilon\rceil$ which yields $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(f(X_t),Y_t)\leq\epsilon$. This holds for any $\epsilon>0$. Therefore, we finally obtain on the event $\mathcal{A}\cap\mathcal{B}\cap\bigcap_{M=1}^{\infty}\mathcal{E}_M$ of probability one, one has $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell(\hat{Y}_t,Y_t)-\ell(f(X_t),Y_t)\leq 0$. This ends the proof that Algorithm 4.6 is universally consistent under CS processes for adversarial empirically integrable responses. Now because there exists a ball

$B_\ell(y, r)$ of $(\mathcal{Y}, \ell)$ that does not satisfy F-TiME, from Theorem 4.11, universal learning with responses restricted on this ball cannot be achieved for processes $\mathbb{X} \notin$ CS. However, these responses are empirically integrable because they are bounded. Hence, CS is still necessary for universal learning with adversarial empirically integrable responses. Thus SOLAR = CS and the provided learning rule is optimistically universal. This ends the proof of the theorem.

4.12.5 Proof of Theorem 4.2

Fix $(\mathcal{X}, \rho_{\mathcal{X}})$ and a value space $(\mathcal{Y}, \ell)$ such that any ball satisfies F-TiME We now construct our learning rule. Let $\bar{y} \in \mathcal{Y}$ be an arbitrary value. For any $M \geq 1$, because $B_\ell(\bar{y}, M)$ is bounded and satisfies F-TiME, there exists an optimistically universal learning rule $f_\cdot^M$ for value space $(B_\ell(y_0, M), \ell)$. For any $M \geq 1$, we define the function $\phi_M : \mathcal{Y} \to \mathcal{Y}$ defined by restricting the space to the ball $B_\ell(\bar{y}, M)$ as follows

$$\phi_M(y) := \begin{cases} y & \text{if } \ell(y, \bar{y}) < M \\ \bar{y} & \text{otherwise.} \end{cases}$$

For simplicity, we will denote by $\hat{Y}_t^M := f_t^M(\mathbb{X}_{\leq t-1}, \phi_M(\mathbb{Y})_{\leq t-1}, X_t)$ the prediction of $f_\cdot^M$ at time t for the responses which are restricted to the ball $B_\ell(\bar{y}, M)$. We now combine these predictors using online learning into a final learning rule $f_\cdot$. Specifically, we define $I_t := \{0 \leq M \leq s^\epsilon \ln t\}$ for all $t \geq 1$. We also denote $t_M = \lceil e^{(2+c_1^\ell)M} \rceil$ for $M \geq 0$ and pose $\eta_t = \frac{1}{4\sqrt{t}}$. Let $s^\epsilon := 1/(2 + c_1^\ell)$. For any $M \in I_t$, we define

$$L_{t-1,M} := \sum_{s=t_M}^{t-1} \ell(\hat{Y}_s^M, \phi_{s^\epsilon \ln s}(Y_s)).$$

For simplicity, we will denote by $\tilde{\mathbb{Y}}$ the process defined by $\tilde{Y}_t = \phi_{s^\epsilon \ln t}(Y_t)$ for all $t \geq 1$. We now construct recursive weights as $w_{0,0} = 1$ and for $t \geq 2$ we pose for all $1 \leq s \leq t-1$

$$\hat{l}_s := \frac{\sum_{M \in I_s} w_{s-1,M} \ell(\hat{Y}_s^M, \tilde{Y}_s)}{\sum_{M \in I_s} w_{s-1,M}}.$$

Now for any $M \in I_t$ we note $\hat{L}_{t-1,M} := \sum_{s=t_M}^{t-1} \hat{\ell}_s$, and pose $w_{t-1,M} := e^{\eta_t(\hat{L}_{t-1,M} - L_{t-1,M})}$. We then choose a random index $\hat{M}_t$ independent from the past history such that

$$\mathbb{P}(\hat{M}_t = M) := \frac{w_{t-1,M}}{\sum_{M' \in I_t} w_{t-1,M'}}, \quad M \in I_t.$$

The output the learning rule is $f_t(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t) := \hat{Y}_t^{\hat{M}_t}$. For simplicity, we will denote by $\hat{Y}_t := f_t(\mathbb{X}_{\leq t-1}, \mathbb{Y}_{\leq t-1}, X_t)$ the prediction of $f_\cdot$ at time t. This ends the construction of our learning rule which is summarized in Algorithm 4.7.

Now let $(\mathbb{X}, \mathbb{Y})$ be such that $\mathbb{X} \in$ SOUL and $\mathbb{Y}$ empirically integrable. By Lemma 4.5, there exists some value $y_0 \in \mathcal{Y}$ such that on an event $\mathcal{A}$ of probability one, we have for any ϵ, a threshold $M_\epsilon \geq 0$ with $\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \ell(y_0, Y_t) \mathbb{1}_{\ell(y_0, Y_t) \geq M_\epsilon} \leq \epsilon$. We fix a measurable

Input: Historical samples $(X_t, Y_t)_{t<T}$ and new input point X_T

Optimistically universal learning rule $f_\cdot^M$ for value space $B_\ell(y_0, M), \ell)$, where $y_0 \in \mathcal{Y}$ fixed.

Output: Predictions $\hat{Y}_t$ for $t \le T$

$I_t := \{0 \le M \le s^\epsilon \ln t\}, \eta_t := \frac{1}{4\sqrt{t}}, t \ge 1$

$t_M = \lceil e^{(2+c_1^\ell)M} \rceil, M \ge 0$

$w_{0,0} := 1, \quad \hat{Y}_1 = y^0 (= f^0(X_0))$ `// Initialisation`

for $t = 2, \dots, T$ **do**

$L_{t-1,M} = \sum_{s=t_M}^{t-1} \ell(f_s^M(\mathbb{X}_{\le s-1}, \phi_M(\mathbb{Y})_{\le s-1}, X_s), \phi_{s^\epsilon \ln s}(Y_s)), \quad \hat{L}_{t-1,M} = \sum_{s=t_M}^{t-1} \hat{\ell}_s, \quad M \in I_t$

$w_{t-1,M} := \exp(\eta_t(\hat{L}_{t-1,M} - L_{t-1,M})), \quad M \in I_t$

$p_t(M) = \frac{w_{t-1,M}}{\sum_{M' \in I_t} w_{t-1,M'}}, \quad M \in I_t$

$\hat{M}_t \sim p_t(\cdot)$ `// Model selection`

$\hat{Y}_t = f_t^{\hat{M}_t}(\mathbb{X}_{\le t-1}, \phi_M(\mathbb{Y})_{\le t-1}, X_t)$

$\hat{\ell}_t := \frac{\sum_{j \in I_t} w_{t-1,j} \ell(f_t^M(\mathbb{X}_{\le t-1}, \phi_M(\mathbb{Y})_{\le t-1}, X_t), \phi_{c_1^\ell \ln t}(Y_t)}{\sum_{j \in I_t} w_{t-1,j}}$

end

Algorithm 4.7: A learning rule for adversarial empirically integrable responses under SMV processes for value spaces $(\mathcal{Y}, \ell)$ such that any ball satisfies F-TiME.

function $f : \mathcal{X} \to \mathcal{Y}$. Also, for any $t \ge 1$ and $M \in I_t$ we have $0 \le \ell(\hat{Y}_t^M, \tilde{Y}_t) \le 2\ell(\hat{Y}_t^M, \bar{y}) + c_1^\ell \ell(\tilde{Y}_t, \bar{y}) \le \ln t$. As a result, for any $M, M' \in I_t$ we have $|\ell(\hat{Y}_t^M, \tilde{Y}_t) - \ell(\hat{Y}_t^{M'}, \tilde{Y}_t)| \le \ln t$. Because $|I_t| \le 1 + \ln t$ for all $t \ge 1$, the same proof as Theorem 4.4 shows that on an event $\mathcal{B}$ of probability one, there exists $\hat{t} \ge 0$ such that

$$\forall t \ge \hat{t}, \forall M \in I_t, \quad \sum_{s=t_M}^{t} \ell(\hat{Y}_t, \tilde{Y}_t) \le \sum_{s=t_M}^{t} \ell(\hat{Y}_t^M, \tilde{Y}_t) + 3 \ln^2 t \sqrt{t}.$$

Further, we know that $f_\cdot^M$ is Bayes optimistically universal for value space $(B_\ell(\bar{y}, M), \ell)$. In particular, because $\mathbb{X} \in \textsc{Soul}$ and $\phi_M \circ f : \mathcal{X} \to B_\ell(\bar{y}, M)$, we have

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t^M, \phi_M(Y_t)) - \ell(\phi_M \circ f(X_t), \phi_M(Y_t)) \le 0 \quad (a.s.).$$

For simplicity, we introduce $\delta_T^M := \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t^M, \phi_M(Y_t)) - \ell(\phi_M \circ f(X_t), \phi_M(Y_t))$ and define $\mathcal{E}_M$ as the event of probability one where the above inequality is satisfied, i.e.,

$\limsup_{T\to\infty} \delta_T^M \le 0$. Because we always have $\ell(\hat{Y}_t, \bar{y}) \le s^\epsilon \ln t$, we can write

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(\hat{Y}_t, \tilde{Y}_t) &= \frac{1}{T}\sum_{t=1}^{T} \Big(\ell(\hat{Y}_t, Y_t) - \ell(\hat{Y}_t, \bar{y})\Big) \mathbb{1}_{\ell(Y_t,\bar{y})\ge s^\epsilon \ln t} \\
&\le \frac{1}{T}\sum_{t=1}^{T} \Big(2\ell(\hat{Y}_t, \bar{y}) + c_1^\ell \ell(Y_t, \bar{y})\Big) \mathbb{1}_{\ell(Y_t,\bar{y})\ge s^\epsilon \ln t} \\
&\le \frac{2 + c_1^\ell}{T}\sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y})\ge s^\epsilon \ln t}.
\end{aligned}$$

The proof of Theorem 4.3 shows that on the event $\mathcal{A}$,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y})\ge s^\epsilon \ln t} \le 0,$$

which implies $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(\hat{Y}_t, \tilde{Y}_t) \le 0$. Now let $M \ge 1$. We write

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t^M, \tilde{Y}_t) - \ell(\hat{Y}_t^M, \phi_M(Y_t)) \\
&\quad\le \frac{1}{T}\sum_{t=1}^{t_M-1} \ell(\hat{Y}_t^M, \tilde{Y}_t) + \frac{1}{T}\sum_{t=t_M}^{T} \Big(\ell(\hat{Y}_t^M, Y_t) - \ell(\hat{Y}_t^M, \bar{y})\Big) \mathbb{1}_{M\le\ell(Y_t,\bar{y})<s^\epsilon \ln t} \\
&\quad\le \frac{t_M(2M + c_1^\ell s^\epsilon \ln t_M)}{T} + \frac{1}{T}\sum_{t=1}^{T} \Big(2\ell(\hat{Y}_t^M, \bar{y}) + c_1^\ell \ell(Y_t, \bar{y})\Big) \mathbb{1}_{\ell(Y_t,\bar{y})\ge M} \\
&\quad\le \frac{t_M(2M + c_1^\ell s^\epsilon \ln t_M)}{T} + \frac{2 + c_1^\ell}{T}\sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y})\ge M}.
\end{aligned}$$

Hence, we obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{Y}_t^M, \tilde{Y}_t) - \ell(\hat{Y}_t^M, \phi_M(Y_t)) \le \limsup_{T\to\infty} \frac{2 + c_1^\ell}{T}\sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y})\ge M}.$$

Finally, we compute

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\ell(\phi_M\circ f(X_t),\phi_M(Y_t))-\ell(f(X_t),Y_t)\\
&\le \frac{1}{T}\sum_{t=1}^{T}\left(\ell(\bar y,Y_t)-\ell(f(X_t),Y_t)\right)\mathbb{1}_{\ell(f(X_t),\bar y)\ge M}\mathbb{1}_{\ell(Y_t,\bar y)\le M}\\
&\qquad+\frac{1}{T}\sum_{t=1}^{T}\left(\ell(f(X_t),\bar y)-\ell(f(X_t),Y_t)\right)\mathbb{1}_{\ell(f(X_t),\bar y)\le M}\mathbb{1}_{\ell(Y_t,\bar y)\ge M}\\
&\le \frac{1}{T}\sum_{t=1}^{T}\ell(\bar y,Y_t)\mathbb{1}_{\ell(Y_t,\bar y)\ge M/(2+c_1^\ell)}+\frac{M}{T}\sum_{t=1}^{T}\mathbb{1}_{\ell(Y_t,\bar y)\ge M}\\
&\qquad+\frac{1}{T}\sum_{t=1}^{T}\left(\ell(\bar y,Y_t)-\ell(f(X_t),Y_t)\right)\mathbb{1}_{\ell(f(X_t),\bar y)\ge M}\mathbb{1}_{\ell(Y_t,\bar y)\le M/(2+c_1^\ell)}\\
&\le \frac{1}{T}\sum_{t=1}^{T}\ell(\bar y,Y_t)\mathbb{1}_{\ell(Y_t,\bar y)\ge M/(2+c_1^\ell)}+\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,\bar y)\mathbb{1}_{\ell(Y_t,\bar y)\ge M}\\
&\qquad+\frac{1}{T}\sum_{t=1}^{T}\left((1+c_1^\ell/2)\ell(\bar y,Y_t)-\ell(f(X_t),\bar y)/2\right)\mathbb{1}_{\ell(f(X_t),\bar y)\ge M}\mathbb{1}_{\ell(Y_t,\bar y)\le M/(2+c_1^\ell)}\\
&\le \frac{1}{T}\sum_{t=1}^{T}\ell(\bar y,Y_t)\mathbb{1}_{\ell(Y_t,\bar y)\ge M/(2+c_1^\ell)}+\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,\bar y)\mathbb{1}_{\ell(Y_t,\bar y)\ge M}.
\end{aligned}
$$

We now put all these estimates together. On the event $\mathcal{A}\cap\mathcal{B}\cap\bigcap_{M=1}^{\infty}\mathcal{E}_M$, for any $M\ge 1$ and $t\ge\max(\hat t,t_M)$ we can write

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\ell(\hat Y_t,Y_t)-\ell(f(X_t),Y_t)\le\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat Y_t,Y_t)-\ell(\hat Y_t,\tilde Y_t)\Big)\\
&+\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat Y_t,\tilde Y_t)-\ell(\hat Y_t^M,\tilde Y_t)\Big)+\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat Y_t^M,\tilde Y_t)-\ell(\hat Y_t^M,\phi_M(Y_t))\Big)+\delta_T^M\\
&+\frac{1}{T}\sum_{t=1}^{T}\left(\ell(\phi_M\circ f(X_t),\phi_M(Y_t))-\ell(f(X_t),Y_t)\right)\\
&\le\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat Y_t,Y_t)-\ell(\hat Y_t,\tilde Y_t)\Big)+\frac{3\ln^2 T}{\sqrt{T}}+\frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\hat Y_t^M,\tilde Y_t)-\ell(\hat Y_t^M,\phi_M(Y_t))\Big)\\
&\qquad+\delta_T^M+\frac{1}{T}\sum_{t=1}^{T}\left(\ell(\phi_M\circ f(X_t),\phi_M(Y_t))-\ell(f(X_t),Y_t)\right).
\end{aligned}
$$

Thus, we obtain on the event $\mathcal{A} \cap \mathcal{B} \cap \bigcap_{M=1}^{\infty} \mathcal{E}_M$, for any $M \geq 1$,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\bar{y}, Y_t) \mathbb{1}_{\ell(Y_t,\bar{y}) \geq M/(2+c_1^\ell)} + (3 + c_1^\ell) \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y}) \geq M}$$

On the event $\mathcal{A}$, the same arguments as in the proof of Theorem 4.3 show that we have same guarantees for y_0 as for $\bar{y}$, i.e., for any $\epsilon > 0$, there exists $\tilde{M}_\epsilon$ such that

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(Y_t, \bar{y}) \mathbb{1}_{\ell(Y_t,\bar{y}) \geq \tilde{M}_\epsilon} \leq \epsilon.$$

Therefore, for any $\epsilon > 0$, we can apply the above equation to $M := (2 + c_1^\ell) M_\epsilon + M_{\epsilon/(3+c_1^\ell)}$ to obtain

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 2\epsilon.$$

Because this holds for all $\epsilon > 0$, we can in finally get

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Big(\ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \Big) \leq 0,$$

on the event $\mathcal{A} \cap \mathcal{E} \cap \bigcap_{M\geq 1} \mathcal{F}_M$ of probability one. This ends the proof of the theorem.

Chapter 5

Contextual Bandits and Optimistically Universal Learning

5.1 Introduction

The contextual bandit setting is one of the core problems in sequential statistical decision-making. Abstractly, in this setting, a learner (or decision maker) interacts with a reward mechanism iteratively. At each iteration, the learner observes a *context* (or covariate vector) $x \in \mathcal{X}$, selects an *arm* (or action) $a \in \mathcal{A}$ to perform, then receives a (potentially stochastic) reward depending on the context and selected action. For example, a store may serve a sequence of customers, for each provide a list of product recommendations, and receive a reward if the recommendation leads to a purchase. The key distinctions between the contextual bandit setting and standard supervised learning (or regression) are that (1) the learner's objective is to obtain a near-maximum average reward over time (rather than merely estimating the reward conditional means), and (2) the learner only observes the reward corresponding to the arm it chose. These aspects introduce a fundamental trade-off between *exploration* and *exploitation*. That is, while some arms may have high estimated reward values, other arms may have higher uncertainty in their rewards and in particular uncertainty about whether they would yield an even higher reward: selecting such arms may provide information about the potential for higher future rewards.

Universal consistency. In the contextual bandit setting, a learner is *consistent* if it has sublinear regret compared to the best policy in hindsight, or equivalently if its average reward converges to the maximum-possible average reward obtained with an optimal policy. Formally, if $\hat{a}_t$ denotes the action selected at time t, an algorithm is consistent if for any policy $\pi^\star : \mathcal{X} \to \mathcal{A}$,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^\star(X_t)) - r_t(\hat{a}_t) \leq 0, \quad (a.s.),$$

where $r_t(a)$ is the reward obtained at time t by selecting action a, and $\mathbb{X} = (X_t)_{t\geq 1}$ is the context sequence. Naturally, one aims for learning procedures consistent under a broad

class of problem instances. In this chapter, we consider the strongest form of consistency: *universal consistency* which asks that a learning rule achieves consistency for any underlying mechanism generating $(r_t)_{t\geq 1}$.

The equivalent notion can be defined for the full-feedback case which we studied in Chapters 3 and 4: for a stream of data $(\mathbb{X}, \mathbb{Y}) = (X_t, Y_t)_{t\geq 1}$ of instances modeled as a stochastic process on $\mathcal{X} \times \mathcal{Y}$, a learning rule with predictions $\hat{Y}_t$ is consistent if it has vanishing excess error compared to any fixed measurable predictor function $f : \mathcal{X} \to \mathcal{Y}$, i.e., $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(\hat{Y}_t, Y_t) - \ell(f(X_t), Y_t) \leq 0$ $(a.s.)$. An algorithm is universally consistent if it is consistent irrespective of the generating process for the values $\mathbb{Y}$ from the instances $\mathbb{X}$. In this standard full-feedback setting, many works established universal consistency, starting with [Sto77] who proved universal consistency for a broad family of *local average estimators*. We refer to Chapter 2 for a thorough literature review. In the previous sections we showed that we can characterize provably-minimal assumptions for this setting using the *optimistically universal* framework [Han21a], succinctly summarized as "learning whenever learning is possible", which we briefly recall here.

Optimistically universal learning. The idea is to identify the minimal assumption on the data sequence $\mathbb{X}$ sufficient for universal consistency to be possible. Such an assumption is both necessary and sufficient, and therefore amounts to merely assuming that universally consistent learning is *possible*: aptly named the *optimist's assumption*. For any process $\mathbb{X}$ satisfying this minimal assumption, by definition there must exist a universally consistent learning rule possibly dependent on $\mathbb{X}$. The interesting question is whether the optimist's assumption alone is sufficient: that is, whether there exists a single learning rule that is universally consistent for *every* process $\mathbb{X}$ satisfying the optimist's assumption. Such a learning rule is said to be *optimistically universal*.

The study of optimistic universal learning for full-feedback proved particularly fruitful as we showed in Chapters 3 and 4. In particular, we can characterize the class of universally learnable processes, which is under mild assumptions very general (beyond i.i.d. or stationary ergodic processes). Further, we were always able to provide optimistically universal learning rules.

Universal learning with partial feedback. The contextual-bandit formulation was first introduced for one-armed bandits [Woo79; Sar91] in a rather restricted setting. Since then, progress has been made to investigate stochastic contextual bandits under *parametric* assumptions [WKP05; LZ07; GZ09; BC+12; AC16; RS16]. In the *non-parametric* setting, advances have been made to obtain minimax guarantees under smoothness conditions and margin assumptions on rewards [LPP09; RZ10; Sli11; PR13; GJ18; RMB18].

However, to the best of our knowledge, no prior works establish universal consistency even under all i.i.d. data sequences, i.e., consistency in the non-parametric setting without further assumptions. As such, the present work is also the first to propose such results and corresponding universally consistent learning rules. Closest to this work is the result from [YZ02] which shows that if rewards are continuous in the contexts, strong consistency can be achieved with familiar non-parametric methods, for Euclidean context spaces. Our work significantly generalizes this result to unrestricted reward mechanisms, separable metric

action and context spaces, and non-i.i.d. data.

Non-i.i.d. context data has also been widely studied in the literature. Examples include customers' profile distribution, which may change depending on seasonal patterns, or the extension of clinical trials to new populations. In these cases, the distribution of contexts x changes while the underlying conditional distribution remains unchanged, a phenomenon known as *covariate-shift*. Such formalism was adopted in works on domain adaptation for classification [Sug+07; Gre+09; BU12]. Moreover, several works have also considered distributional shifts in both contexts and responses for bandit problems, in both parametric [BGZ14; Luo+18; WIW18; Che+19] and non-parametric settings [SK21].

5.1.1 Summary of the chapter

We study optimistically universal learning in standard contextual bandits [Sli+19; LS20]. Precisely, we assume that there exists a time-invariant conditional probability distribution $P_{r|a,x}$ such that the reward r_t is sampled according to $P_{r|a=a_t,x=X_t}$ where a_t (resp. X_t) is the selected action (resp. observed context) at time t, independently from the past history. We are interested in online learning, where the learner may observe all past rewards $r_{t'}$ and contexts $X_{t'}$, $t' < t$, when choosing its action a_t given the context X_t. We aim to achieve average reward $\frac{1}{T}\sum_{t=1}^{T} r_t$ that is competitive with any fixed policy $\mathcal{X} \to \mathcal{A}$ as $T \to \infty$. Our results show that consistency is achievable under large classes of context processes and without assumptions about the probability distribution $P_{r|a,x}$. In particular, to the best of our knowledge, this is the first work giving algorithms consistent under any such contextual bandit instance with finite action spaces and i.i.d. contexts. Our results go well beyond this standard setting and are summarized below.

Bounded unrestricted rewards We first focus on the classical assumption of bounded rewards and show there always exists an optimistically universal learning rule. Our approach is to first characterize which processes $\mathbb{X}$ admit universally consistent learning rules, then use this characterization to inform the design of a learning rule, which will be universally consistent under every such process. This turns out to require three separate cases: $\mathcal{A}$ finite, $\mathcal{A}$ countably infinite, and $\mathcal{A}$ uncountably infinite. Each of these cases gives rise to a different characterization of the set of processes $\mathbb{X}$ under which universally consistent learning is possible for contextual bandits, a fact which itself is of independent interest. Moreover, each of these sets of processes corresponds to known families of processes from the past literature on optimistically universal learning. When $\mathcal{A}$ is finite, the set of processes admitting universal consistency for contextual bandits is equivalent to the family of processes admitting universally consistent online learning with full supervision: Condition SMV. While this appears natural, interestingly this is not the case when $\mathcal{A}$ is countably infinite. In that case, the set of processes admitting universal learning for contextual bandits is equivalent to the family of processes admitting universally consistent *inductive* learning with full supervision: Condition CS, which is more restrictive than SMV. Finally, when $\mathcal{A}$ is uncountably infinite, universal learning can never be achieved. We note that both CS and SMV are very general classes of processes, encompassing in particular i.i.d., ergodic, and stationary processes.

Bounded rewards under continuity assumptions For unrestricted rewards, although large classes of non-i.i.d. processes (CS or SMV) admit universal learning for countable action spaces, the answer for uncountable action spaces is disappointing: universal consistency is impossible. However, we show that under continuity assumptions on the rewards, one can recover positive results for general action spaces. Further, in all cases, we provide optimistically universal learning rules. First, under the assumption that rewards are continuous, the characterization of processes admitting universal consistency now requires only two cases. If the action space is finite, the set of processes admitting universal learning remains unchanged and is SMV. On the other hand, if the action space is infinite, this set becomes CS, irrespective of whether the action space was countably or uncountably infinite. Second, we consider a stronger assumption of uniform continuity on the rewards, in which the modulus of continuity of the expected reward in the actions $\bar{r}(\cdot, x)$ for $x \in \mathcal{X}$ are uniform over the context space $\mathcal{X}$. Under this assumption, universal learning under the more general set of processes SMV becomes possible for a significantly larger class of action spaces, namely totally-bounded action spaces. Otherwise, universal learning is achievable exactly on CS processes.

Unbounded rewards Last, we characterize and give optimistically universal learning rules the most general case of unbounded rewards. Even with full supervision, universal consistency for *unbounded* losses is known to be very restrictive. This is possible only for processes visiting only a finite number of distinct instances in $\mathcal{X}$: Condition FS. For contextual bandits, in the standard case of unrestricted rewards, we show that there is a simple dichotomy: if the action space is countable then the set of processes admitting universal learning is still FS; however, if the action space is uncountably infinite, universal learning can never be achieved. Nevertheless, under continuity assumptions on the rewards, universal learning can always be achieved under FS processes.

5.1.2 Overview of probability-theoretic contributions

In this chapter, we make use of the conditions CS, SMV, and FS on stochastic processes from the universal learning literature to characterize the set of processes admitting universal learning. Along the way to establishing these results, another significant contribution of this chapter is establishing new equivalent characterizations of the families CS and SMV, crucial for the design of our optimistically universal algorithms. In particular, we establish a new connection between these two families: proving that SMV can essentially be characterized by processes that would be in CS if we were to replace duplicate values in the sequence $\mathbb{X}$ by some default value x_0. As a result, SMV processes differ from CS processes only through duplicates: if a process $\mathbb{X}$ is guaranteed to never visit the same context with probability one (e.g. i.i.d. processes with density) the properties CS and SMV are equivalent. This fact has further interesting implications, such as a new technique for the design of optimistically universal learning rules for online learning with full supervision; this gives alternative optimistically universal learning rules to the modified nearest neighbor algorithm 2C1NN that we introduced in Chapter 3. The new approach suggested in the present chapter is instead based on model selection techniques, in the spirit of structural risk minimization [Han21a].

5.1.3 Overview of algorithmic techniques

We present an overview of the optimistically universal learning rule for finite action sets, Algorithm 5.5, which encompasses the main algorithmic innovation in this chapter. We use the property that SMV processes without duplicates satisfy the CS property (Proposition 5.1) to separate times into two classes: points not appearing often recently and points which have many duplicates recently.

1. For the points in the first category, which behave as CS processes, we use an approach similar to structural risk minimization: we aim to achieve sublinear regret compared to a constructed countable set of policies that is empirically dense. To do so, we use a restarting technique introduced in [Han21a]: we use classical bandit algorithms as a subroutine to achieve sublinear regret compared to a fixed finite number of policies, and occasionally restart the bandit learner to gradually increase the number of competing policies considered.

2. For the points in the second category, we use a completely different strategy. Intuitively, these correspond to instances with many duplicates in the recent past, hence it is advantageous to assign each frequent instance an independent bandit learner. In particular, this specific bandit learner is tailored to that point's rewards only and completely disregards historical data from other points.

Interestingly, we can interpret the general strategy as balancing a trade-off between *generalization* and *personalization*. The first strategy aims to find a policy that performs well at an aggregate level for points with few duplicates. On the other hand, the algorithm performs pure personalization for specific points that have many recent repetitions. This schematic presentation hides many details. To obtain vanishing excess error compared to the optimal policy, the algorithm needs to balance the generalization/personalization trade-off carefully. In effect, we allow for a cap of M of duplicates for each instance in the recent past to be treated with the generalization strategy and adaptively increase this cap. To adaptively increase this cap, the algorithm occasionally uses "exploration" times to estimate the performance of each strategy, and decides to increase the cap based on these estimates. Last, to have decisions robust to non-stationarity in the sequence of contexts, the algorithm selects actions based on recent data: the learning procedure is broken down by periods that contain a given proportion of the past data, and this proportion adaptively decays to 0.

5.1.4 Outline of the chapter

After giving the definitions and main results in Section 5.2, we provide in Section 5.3 new characterizations of stochastic process classes as well as base algorithms, used to construct our learning rules. With these tools, we study optimistic learning with bounded rewards for finite (Section 5.4) countably infinite (Section 5.5), and uncountable (Section 5.6) action sets. We then show that universal learning can be achieved on larger classes of processes under continuity assumptions on the rewards in Section 5.7. Last, we treat the more restrictive case of unbounded rewards in Section 5.8 and give remaining proofs in the appendix Section 5.9.

5.2 Preliminaries and Main Results

5.2.1 Formal setup and problem formulation

The goal of this chapter is to study the general framework of contextual bandits in an online setting. Given a separable metrizable Borel context space $(\mathcal{X}, \mathcal{B})$ and a separable metrizable Borel action space $\mathcal{A}$, the learner interacts with the contextual bandit at each iteration $t \geq 1$ of the learning process in the following fashion. First, the learner observes a context $X_t \in \mathcal{X}$, then selects an action $\hat{a}_t \in \mathcal{A}$ based on the past history only. As a result of the action, the learner receives a reward r_t. We will suppose for the most part that the rewards are bounded $r_t \in [0, R] = \mathcal{R}$ for some known $R \geq 0$. Without loss of generality we take $R = 1$. Crucially, the learning rule can only use the past history, as defined below.

Definition 5.1 (Learning rule)**.** *A* learning rule *is a sequence $f_\cdot = (f_t)_{t\geq 1}$ of possibly randomized measurable functions $f_t : \mathcal{X}^{t-1} \times \mathcal{R}^{t-1} \times \mathcal{X} \to \mathcal{A}$. The action selected at time t by the learning rule is $\hat{a}_t = f_t((X_s)_{s\leq t-1}, (r_s)_{s\leq t-1}, X_t)$.*

We now describe the standard contextual bandit model. We suppose that the contexts are generated from a stochastic process $\mathbb{X} = (X_t)_{t\in\mathbb{N}}$ on $\mathcal{X}$. Further, we assume that rewards are sampled from a distribution conditionally on the context and actions. Formally, there is a time-invariant conditional distribution $P_{r|a,x}$ such that the rewards $(r_t)_{t\geq 1}$ are conditionally independent given their respective selected action a_t and observed context x_t, and follow this conditional distribution: $(r_t \mid a_t, x_t)_{t\geq 1} \overset{iid.}{\sim} P_{r|a,x}$. To emphasize the conditional dependence of r_t on the actions and context, we denote $r_t(a, x)$ (resp. $r_t(a)$) the reward at time t, had the selected action been $a \in \mathcal{A}$ and the observed context $x \in \mathcal{X}$ (resp. when the context at time t is clear). Further, by abuse of notation, we will refer to a reward mechanism r as a random variable $r \sim P_{r|a,x}$. We use the notation $\bar{r}(a, x) = \mathbb{E}[r \mid a, x]$ to denote the expected reward for any $a \in \mathcal{A}$ and $x \in \mathcal{X}$. For unbounded rewards, we assume that the random variable $r(a, x)$ is integrable for any $(a, x) \in \mathcal{A} \times \mathcal{X}$. We consider three settings for the reward mechanism r: unrestricted, continuous, and uniformly-continuous. For the two last settings defined below, we suppose that $\mathcal{A}$ is a separable metric space with metric d.

Definition 5.2. *The reward mechanism r is* continuous *if for any $x \in \mathcal{X}$, the immediate expected reward function $\bar{r}(\cdot, x) : \mathcal{A} \to [0, 1]$ is continuous.*
The reward mechanism r is uniformly-continuous *if for any $\epsilon > 0$ there exists $\Delta(\epsilon) > 0$ with*

$$\forall x \in \mathcal{X}, \forall a, a' \in \mathcal{A}, \quad d(a, a') \leq \Delta(\epsilon) \Rightarrow |\bar{r}(a, x) - \bar{r}(a', x)| \leq \epsilon.$$

Our goal is to design algorithms that intuitively converge to the optimal policy $\pi^* : \mathcal{X} \to \mathcal{A}$ that selects for any context $x \in \mathcal{X}$ an optimal arm in $\arg\max_{a\in\mathcal{A}} \bar{r}(a, x)$. Such an optimal policy π^* is well-defined for finite $\mathcal{A}$; however, for infinite $\mathcal{A}$, this may no longer be the case. Instead, to be fully general, we ask that the regret of the algorithm be sublinear compared to *any* fixed measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$. We are then interested in learning rules that are consistent irrespective of the unknown reward mechanism r, i.e., that always converges to a (near-)optimal policy.

Definition 5.3 (Consistency and universal consistency)**.** *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$, r a reward mechanism, and f. a learning rule. Denote by $(\hat{a}_t)_{t\geq 1}$ it's selected actions. We say that f. is* consistent *under $\mathbb{X}$ with rewards r if for any measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq 0, \quad (a.s.).$$

The rule f. is universally consistent *under $\mathbb{X}$ if it is consistent for any reward mechanism r.*

Intuitively, universal learning rules are the most general possible: standard approaches in contextual bandits consider a specific family of reward functions for which they obtain guarantees, while in universal learning, the set of rewards considered is unrestricted. This is of course a very strong objective and unfortunately, universal consistency is not always achievable. For a simple example, on $\mathcal{X} = \mathbb{N}$, under the process $\mathbb{X} = (t)_{t\geq 1}$, there does not exist any universally consistent learning rule because the best action in hindsight at the context t may be completely unrelated to that from previously observed contexts $t' < t$. Two natural questions then arise. First, for which stochastic processes is universal consistency possible? And second, which algorithms are universally consistent for all such stochastic processes? This latter property is particularly appealing, as it means the learning rule is provably universally consistent given *only* the assumption that universal consistency is possible under the given process (it "learns whenever learning is possible"); this is the *minimal* assumption on the process under which one could hope to prove universal consistency. Such learning rules are called *optimistically universal*, as defined formally below.

Definition 5.4 (Optimistically universal learning rule)**.** *Let Socb be the set of processes $\mathbb{X}$ on $\mathcal{X}$ such that there exists a learning rule universally consistent under $\mathbb{X}$. We say that a learning rule f. is* optimistically universal *if it is universally consistent under every process $\mathbb{X} \in$ Socb.*

Similarly, let Socb-C (resp. Socb-Uc) be the set of processes admitting universally consistent learning under continuous (resp. uniformly-continuous) rewards and define optimistically universal learning rule for continuous (resp. uniformly-continuous) rewards accordingly.

In this chapter, we answer the questions raised above by (1) characterizing the set of processes Socb admitting universally consistent learners, and (2) proving that there indeed exist optimistically universal learning rules and providing explicit definitions of such learners.

5.2.2 Useful classes of stochastic processes

The conditions that will arise in our universal learning characterizations are the same as those introduced in Chapters 2 to 4. We briefly recall these. Let us first start with some notation. For any stochastic process $\mathbb{X} = (X_t)_{t\geq 1}$, we denote $\mathbb{X}_{\leq t} = (X_s)_{s\leq t}$ for any $t \geq 1$. We also introduce the empirical limsup frequency $\hat{\mu}_{\mathbb{X}}$ via $\hat{\mu}_{\mathbb{X}}(A) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_A(X_t)$ for any $A \in \mathcal{B}$. $\hat{\mu}_{\mathbb{X}}(A)$ quantifies the asymptotic proportion of points falling in A. The first class of processes we defined in Condition CS are those for which $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(\cdot)]$ forms a *continuous sub-measure*.

Condition CS. *For every decreasing sequence* $\{A_k\}_{k=1}^{\infty}$ *of measurable sets in* $\mathcal{X}$ *with* $A_k \downarrow \emptyset$, $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_k)] \underset{k\to\infty}{\longrightarrow} 0$.

This family CS of processes is very large and includes i.i.d. processes, all stationary processes, and in fact all processes satisfying the law of large numbers— that is, for any $A \in \mathcal{B}$, the limit $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_A(X_t)$ exists almost surely [Han21a]. It also includes many non-stationary processes [Han21a]. Algorithmic details on how this condition is useful for learning are deferred to Section 5.3.2. We next introduce an even more general class of processes, Condition SMV which asks that the process visits a sublinear number of sets from any countable measurable partition of $\mathcal{X}$.

Condition SMV. *For every disjoint sequence* $\{A_k\}_{k=1}^{\infty}$ *of measurable sets of* $\mathcal{X}$ *such that* $\bigcup_{k=1}^{\infty} A_k = \mathcal{X}$, *(every countable measurable partition),* $|\{k \geq 1 : A_k \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$, $(a.s.)$.

It is known [Han21a] that CS $\subset$ SMV. Therefore, both conditions CS and SMV encompass large classes of processes and generalize standard assumptions on processes as described above. To briefly explain why CS $\subset$ SMV, at a high level, if $\mathbb{X}$ visits new disjoint regions A_k linearly often, then the tail union set $B_k = \bigcup_{l\geq k} A_l$ is visited linearly often, for all k, which violates CS since $B_k \downarrow \emptyset$. The opposite inclusion does *not* hold when $\mathcal{X}$ is infinite [Han21a]: for instance, we may exhibit a deterministic process $(X_l)_{l\geq 1} \in$ SMV $\setminus$ CS taking $X_l = x_{\lceil\sqrt{l}\rceil}$ for any sequence x_i of distinct points. Last, we introduce a significantly smaller class of processes, based on a condition asking that the process only visits a finite number of distinct points.

Condition FS. *The process* $\mathbb{X}$ *satisfies* $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X} \neq \emptyset\}| < \infty$ $(a.s.)$.

This condition is rather restrictive and only appears in our characterizations of universal learning under unbounded rewards. It does not include i.i.d. processes in general: for spaces $\mathcal{X}$ admitting a non-atomic measure μ, an i.i.d. process $\mathbb{X} \overset{i.i.d.}{\sim} \mu$ almost surely does not visit the same point twice (e.g., the uniform distribution on $[0, 1]$). We can also directly check FS $\subset$ CS since for $A_k \downarrow \emptyset$, the condition FS ensures that with probability one, there exists an index $\hat{k}$ such that $A_{\hat{k}}$ is never visited.

5.2.3 Main results

We now present our main results. We show that the set of processes admitting universal learning Socb corresponds to one of the classes FS $\subset$ CS $\subset$ SMV and depends *only* on the action set $\mathcal{A}$. A summary of the characterizations is provided in Table 5.1. We also give optimistically universal learning rules for each case (see Sections 5.4 to 5.7). In the main setting of bounded rewards, the relevant distinctions are whether $\mathcal{A}$ is finite, countably infinite, or uncountable.

Theorem 5.1 (Unrestricted bounded rewards)**.** *Let* $\mathcal{X}$ *be a separable metrizable Borel context space and* $\mathcal{A}$ *an action space. Then,*

- *If* $\mathcal{A}$ *is finite and* $|\mathcal{A}| \geq 2$, *then* Socb $=$ *SMV.*

	Unrestricted rewards	Continuous rewards	Uniformly-continuous rewards
Bounded rewards	Finite: SMV Countably infinite: CS Uncountable: $\emptyset$	Finite: SMV Infinite: CS	Totally-bounded: SMV Non-totally-bounded: CS
Unbounded rewards	Countable: FS Uncountable: $\emptyset$	FS	FS

Table 5.1: Characterization of learnable instance processes for universal learning in contextual bandits depending on properties of the action space $\mathcal{A}$.

- *If $\mathcal{A}$ is countably infinite, then* $\textsc{Socb} = \textsc{CS}$.
- *If $\mathcal{A}$ is an uncountable separable metrizable Borel space, then* $\textsc{Socb} = \emptyset$.

In all cases, there is an optimistically universal learning rule.

The proof spans Section 5.4 for finite, Section 5.5 for countably infinite, and Section 5.6 for uncountable action spaces. Recall that $\mathbb{X} \in \text{SMV}$ is necessary to achieve universal learning under $\mathbb{X}$ even in the simplest online learning setting with full-feedback and noiseless values (see Chapter 3). Therefore, Theorem 5.1 shows that universal consistency for contextual bandits is achievable for finite action sets at no extra cost compared to full-feedback. For countably infinite action spaces, the situation is more nuanced, as we find the more-restrictive condition CS is necessary; the class CS is known to characterize universal learnability for full-feedback in certain variants of the learning setting, such as *inductive* learning [Han21a] and online learning with *adversarial* responses under certain loss functions (see Chapter 4), so that the online contextual bandit problem with countably infinite action spaces is essentially of equivalent difficulty to these. On the other hand, in uncountable action spaces, universal consistency is not achievable. A natural question then becomes whether under mild assumptions on the rewards one can recover the large classes of processes CS or SMV for universal learning. In particular, if we assume that $(\mathcal{A}, d)$ is a separable metric space and first consider the case of *continuous* rewards, we show that the set $\textsc{Socb}$-C of processes admitting universal consistency is equal CS for infinite $\mathcal{A}$.

Theorem 5.2 (Continuous bounded rewards)**.** *Let $\mathcal{X}$ be a separable metrizable Borel context space and $(\mathcal{A}, d)$ a separable metric action space. Then,*

- *If $\mathcal{A}$ is finite and $|\mathcal{A}| \geq 2$, then* $\textsc{Socb}\text{-}C = \text{SMV}$.
- *If $\mathcal{A}$ is infinite, then* $\textsc{Socb}\text{-}C = \textsc{CS}$.

In all cases, there is an optimistically universal learning rule for continuous rewards.

The proof is given in Section 5.7.1. As a result, under the continuity assumption, one recovers the set of processes CS for infinite action spaces. However, it is not sufficient to recover the largest set SMV which is necessary even in the noiseless full-feedback setting. However, if we consider the stronger assumption that rewards are uniformly-continuous, then we show that one can recover the full set of processes SMV for universally consistent learning under all *totally-bounded* action spaces for the class $\textsc{Socb}$-Uc of universally learnable processes for uniformly-continuous rewards.

Theorem 5.3 (Uniformly-continuous bounded rewards)**.** *Let $\mathcal{X}$ be a separable metrizable Borel context space and $(\mathcal{A}, d)$ a separable metric action space. Then,*

- *If $\mathcal{A}$ is totally-bounded and $|\mathcal{A}| \geq 2$, then SOCB-UC = SMV.*
- *If $\mathcal{A}$ is non-totally-bounded, then SOCB-UC = CS.*

In all cases, there is an optimistically universal learning rule for uniformly-continuous rewards.

The proof is given in Section 5.7.2. Last, we consider unbounded rewards in $\mathcal{R} = [0, \infty)$. In Chapter 3, we showed that even in the simplest noiseless full-feedback case, FS is necessary for universal learning with unbounded rewards. We show that universal learning under FS processes is generally still possible for contextual bandits but find that neither continuity nor uniform continuity assumptions are sufficient to extend beyond FS. The proof of the result below is given in Section 5.8.

Theorem 5.4 (Unbounded rewards)**.** *Let $\mathcal{X}$ be a separable metrizable Borel context space and $(\mathcal{A}, d)$ a separable metric action space.*

- *If $\mathcal{A}$ is countable, and $|\mathcal{A}| \geq 2$, then SOCB = FS. If $\mathcal{A}$ is uncountable, then SOCB = $\emptyset$.*
- *For any $\mathcal{A}$ with $|\mathcal{A}| \geq 2$, SOCB-C = SOCB-UC = FS.*

In all cases, there is an optimistically universal learning rule for all the rewards models.

5.3 Base Ingredients for the Proofs and Algorithms

5.3.1 Equivalent characterizations of stochastic process classes

We give new characterizations of the classes CS and SMV, which are central to our proofs, and also of independent interest. We first show that $\mathbb{X} \notin$ CS if and only if we can construct a measurable partition visited linearly by the process up to a known maximum number of duplicated instances for each set of the partition.

Lemma 5.1. *$\mathbb{X} \notin$ CS if and only if the following holds: there exists a disjoint sequence $\{B_i\}_{i=1}^{\infty}$ of measurable subsets of $\mathcal{X}$ with $\bigcup_{i \in \mathbb{N}} B_i = \mathcal{X}$, and a sequence N_i in $\mathbb{N}$ such that, letting i_t be the unique $i \in \mathbb{N}$ with $X_t \in B_i$, with probability strictly greater than zero,*

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] > 0.$$

Intuitively, this characterization shows that for $\mathbb{X} \notin$ CS, there is an infinite number of regions that are each sparsely visited by the process, but still together form a significant proportion of times. Thanks to this property, we show that universal learning beyond CS processes is impossible for infinite action spaces. When choosing an action for a context X_t in a region B_i which was encountered fewer than a bounded number N_i of times before,

the learner is not able to guess an optimal action within an infinite number of possibilities. Hence, one can construct rewards such that this happens for the first N_i visits of B_i, for all $i \geq 1$ with high probability. Because these represent a constant proportion of all times as per Lemma 5.1, the algorithm is not consistent.

Next, we give a new characterization of SMV processes, revealing a fascinating new expression of the relation between CS and SMV. We consider *sparsified* stochastic processes which may take their defined values on a subset of possibly random ($\mathbb{X}$-dependent) times $\mathcal{T} \subset \mathbb{N}$ instead of the complete set of times $\mathbb{N}$, and fill the remaining times with any fixed "dummy" value $x_\varnothing \notin \mathcal{X}$. Specifically, for any process $\mathbb{X} = (X_t)_{t\geq 1}$ and $\mathbb{X}$-dependent random set $\mathcal{T} \subseteq \mathbb{N}$, define the stochastic process $\mathbb{X}^{\mathcal{T}} = (X_t^{\mathcal{T}})_{t\geq 1}$ on $\mathcal{X} \cup \{x_\varnothing\}$ (extending the σ-algebra appropriately) by

$$X_t^{\mathcal{T}} = \begin{cases} X_t & \text{if } t \in \mathcal{T} \\ x_\varnothing & \text{otherwise} \end{cases}.$$

The purpose of this sparsified process is that the times $t \notin \mathcal{T}$ have X_t replaced by a *non-value* $x_\varnothing$, which therefore does not contribute to empirical frequencies in $\hat{\mu}_{\mathbb{X}^{\mathcal{T}}}(A)$ for sets $A \subseteq \mathcal{X}$. This modification leads to an *extended* definition of CS for *sparsified* processes with the same definition as in Condition CS. That is, for a process $\mathbb{X}$ and an $\mathbb{X}$-dependent $\mathcal{T}$, we say that $\mathbb{X}^{\mathcal{T}} \in$ CS if every monotone sequence A_k of measurable subsets of $\mathcal{X}$ with $A_k \downarrow \emptyset$ satisfies $\lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{\mathbb{X}^{\mathcal{T}}}(A_k)] = 0$, or equivalently,

$$\lim_{k\to\infty} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_k}(X_t)\right] = 0.$$

As an important remark, this set of extended CS stochastic processes can be larger than the processes $\mathbb{X}$ with $(X_t)_{t\in\mathcal{T}} \in$ CS. For instance, on $\mathcal{X} = \mathbb{N}$, consider the process $(X_t = t)_{t\geq 1}$, and $\mathcal{T} = \{t_k : k \geq 1\}$ for an increasing sequence t_k with $t_k/k \to \infty$. We can easily check that $\mathbb{X}^{\mathcal{T}} \in$ CS, but the process $(X_{t_k})_{k\geq 1}$ does not belong to CS—the sets $A_k = \{n \geq k\}$ for $k \geq 1$ disprove the condition.

The following result shows that SMV is equivalent to such an extended variant of CS, for appropriate choices of $\mathcal{T}$: namely, those forcing a bounded number of *duplicate* values.

Proposition 5.1. *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$, and for $M \geq 1$, define an $\mathbb{X}$-dependent set*

$$\mathcal{T}^{\leq M} = \left\{t \geq 1 : \sum_{t'\leq t} \mathbb{1}[X_{t'} = X_t] \leq M\right\},$$

the set of times which are duplicates with index at most M. In particular, $\mathcal{T}^{\leq 1}$ is the set of all times of first appearances *of values. Similarly, $\mathcal{T}^{<M} = \mathcal{T}^{\leq M-1}$ for $M \geq 2$. For brevity, we introduce the shorthand notation $\mathbb{X}^{(\leq M)} = \mathbb{X}^{\mathcal{T}^{\leq M}}$. The following are equivalent.*

1. *$\mathbb{X} \in$ SMV.*
2. *$\mathbb{X}^{(\leq 1)} \in$ CS.*

3. For all $M \geq 1$, $\mathbb{X}^{(\leq M)} \in CS$.

In other words, denoting by $\hat{\mu}_{\mathbb{X}}^{(\leq M)}(A) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq M}} \mathbb{1}_A(X_t)$, the result shows that $\mathbb{X} \in$ SMV if and only if $\mathbb{E}[\hat{\mu}_{\mathbb{X}}^{(\leq 1)}(\cdot)]$ is a continuous submeasure. This result implies that the main difference between CS and SMV processes lies in the multiple occurrences of values. This fundamental connection between SMV and CS had not previously been identified in the literature, and in addition to being central to our analysis below, is also of independent interest. In particular, if $\mathbb{X}$ never visits the same value twice almost surely, as is the case of i.i.d. processes with densities, then $\mathbb{X} \in$ CS if and only if $\mathbb{X} \in$ SMV. We give a simple process $\mathbb{X} \in \text{SMV} \setminus \text{CS}$ that exemplifies this distinction. Let $(x_k)_{k\geq 1}$ be a sequence of distinct points in $\mathcal{X}$. Let $\mathbb{X}$ be the deterministic sequence that visits x_1 once, then x_2 twice, etc., so that $X_t = x_k$ with $k(k-1)/2 < t \leq k(k+1)/2$. This process is not in CS because the sets $A_k = \{x_l, l \geq k\}$ are visited with asymptotic rate 1. However, it visits new points from the sequence $(x_k)_{k\geq 1}$ at a sublinear rate, hence $\mathbb{X} \in$ SMV. Intuitively, learning under this process should be possible (for finite action spaces) because each point x_k is visited k times, hence a standard bandit learner assigned to x_k would yield an average regret of the order $1/\sqrt{k}$ on these times. This pure personalization approach can be proved to be universally consistent in this example.

Impact of duplicates for learning contextual bandits. Given that SMV processes become CS if one replaces all duplicates by an arbitrary context $x_\varnothing$, one may wonder why duplicates should add complexity to the problem. On one hand, having duplicates can be helpful for the learner since it has access to more information on a single instance and can therefore personalize actions to this instance. On the other hand, the consistency objective is less forgiving: if the learner does not identify a satisfactory action for a point with many duplicates, the regret incurred is proportional to the number of duplicates. Precisely, the learner faces the following dilemma: either (1) the optimal action for an instance with many duplicates is very distinct from similar instances in $\mathbb{X}$, in which case *personalization* is beneficial; or (2) the optimal action is similar to that for similar instances in $\mathbb{X}$, in which case it is more beneficial to rely on *generalization*, using an action observed to yield high rewards among such similar instances (analogous to familiar learning principles from contextual bandits with i.i.d. contexts). The learner does not know a priori which of these two scenarios a particular instance falls in, and incurs a significant regret if it uses the wrong strategy. Eventually, the algorithm should switch to *generalization* for instances duplicated at most a fixed number M of times, since a personalized bandit incurs a small but non-zero regret (roughly $\sqrt{M}$), which is unacceptable if there are $\Omega(T)$ such instance in the process. On the other hand, for instances duplicated an unbounded number of times, the personalization strategy may sometimes be preferable. Deciding when to switch between personalization and generalization is therefore a crucial challenge in the design of a universally consistent learner under SMV processes.

Implications for the full feedback setting. As a consequence of Proposition 5.1, we obtain new major insights on the noiseless full-feedback setting. In this setting, an online learner sequentially observes an instance $X_t \in \mathcal{X}$, predicts a value $\hat{Y}_t \in \mathcal{Y}$ then observes the true value $Y_t = f^*(X_t)$ for some unknown measurable function $f^* : \mathcal{X} \to \mathcal{Y}$. The goal

is to find learning rules satisfying $\frac{1}{T}\sum_{t=1}^{T}\ell(Y_t,\hat{Y}_t)\to 0 \quad (a.s.)$, where ℓ is a given near-metric on $\mathcal{Y}$ (i.e., a loss function), for any f^*. For this setting, [Han21a] gave an algorithm combining the Hedge algorithm and a "dense" countable family of measurable functions, universally consistent under CS processes. In Chapter 3, we then gave a simple 1-nearest-neighbor-based algorithm named 2C1NN and showed that it is universally consistent under SMV processes, which are also necessary for universal learning [Han21a]. Proposition 5.1 directly implies that combining the original algorithm from [Han21a] on first-appearances of each value X_t, i.e., on times $\mathcal{T}^{\leq 1}$, with memorization for previously observed instances, also yields an optimistically universal learning rule. Unfortunately, such a direct argument does not extend to the setting of noisy responses Y_t from Chapter 4, where the values Y_t may not come from a fixed measurable function $f^*(X_t)$. In such cases, the trade-off between generalization and personalization is again crucial.

5.3.2 Algorithms for learning with experts

We give the main ingredients that will be used as sub-routines in our algorithms. We start by recalling the classic EXP3 algorithm [Aue+02] for multi-armed bandits, and the corresponding guarantee on its regret. The EXP3 algorithm is designed for the general (adversarial) K-armed bandit setting: i.e., where there is no context, and the rewards $r_t(a)\in[0,1]$ at time t for arm $a\in\{a_1,\dots,a_K\}$ are set by an adversary with knowledge of the algorithm and its choices of arms $\hat{a}_1,\dots,\hat{a}_{t-1}$, but without knowledge of the algorithm's randomness regarding its next choice of arm $\hat{a}_t$. This repeats for a total of T rounds, and we are interested in the regret compared to the best fixed choice of arm a_i in hindsight. The EXP3 algorithm initializes a distribution p_1 uniform over $\{a_1,\dots,a_K\}$ and values $L_{a_i}=0$ $(\forall i\leq K)$; for each round $t=1,2,\dots,T$, it samples its choice $\hat{a}_t\sim p_t$, updates $L_{\hat{a}_t}\leftarrow L_{\hat{a}_t}+(1-r_t(\hat{a}_t))/p_t(\hat{a}_t)$, and defines $p_{t+1}(a_i)\propto e^{-\eta_t L_{a_i}}$, where η_t is a prespecified value. The following describes a known performance guarantee for this algorithm.

Theorem 5.5 (Expected regret of EXP3 [BC+12])**.** *If* EXP3 *is run with parameters* $\eta_t=\sqrt{\frac{\ln K}{tK}}$ *on a multi-armed bandit with* K *arms, then it satisfies the following "pseudo-regret" guarantee:*

$$\max_{i\in[K]}\mathbb{E}\left[\sum_{t=1}^{T}r_t(a_i)\right]-\mathbb{E}\left[\sum_{t=1}^{T}r_t(\hat{a}_t)\right]\leq 2\sqrt{TK\ln K}.$$

We will also need an algorithm for adversarial multi-armed bandits guaranteeing a regret bound holding with high probability $1-\delta$, and moreover having no explicit dependence on δ or T. Such an algorithm, called EXP3.IX, was proposed by [Neu15]. The algorithm is identical to the description of EXP3 above, except that the update to $L_{\hat{a}_t}$ is now given as $L_{\hat{a}_t}\leftarrow L_{\hat{a}_t}+(1-r_t(\hat{a}_t))/(p_t(\hat{a}_t)+\gamma_t)$, where γ_t is a prespecified value. [Neu15] establishes the following result, taking $\eta_t=2\gamma_t=\sqrt{\frac{\ln K}{tK}}$.

Theorem 5.6 (High-probability regret of EXP3.IX [Neu15])**.** *For adversarial bandits with* K *arms,* EXP3.IX *satisfies that, for any* $\delta\in(0,1)$ *and* $T\geq 1$*, with probability at least* $1-\delta$*,*

$$\max_{i\in[K]}\sum_{t=1}^{T}(r_t(a_i)-r_t(\hat{a}_t))\leq 4\sqrt{KT\ln K}+\left(2\sqrt{\frac{KT}{\ln K}}+1\right)\ln\frac{2}{\delta}.$$

For our purposes, it will always suffice to use a simplified version of this result: there exists a universal constant $c > 0$ such that, for any $\delta \in (0, 1/2]$, with probability at least $1 - \delta$,

$$\max_{i \in [K]} \sum_{t=1}^{T} (r_t(a_i) - r_t(\hat{a}_t)) \leq c\sqrt{KT \ln K} \ln \frac{1}{\delta},$$

This has the following corollary which allows one to consider a countable family of experts asymptotically, based on an argument from [Han22, Corollary 4]. We use the same construction to design an algorithm EXPINF for learning with a countably infinite number of "experts" (where an *expert* E_i, in this context, provides a *suggested action* $E_{i,t} \in \mathcal{A}$ on each round t, before the learner chooses its action $\hat{a}_t$). The original proof of [Han22] extended the Hedge algorithm [Ces+97a] to an infinite number of experts in the full-feedback setting, but the argument remains valid (with only superficial changes) when applied with EXP3.IX in the bandit-feedback setting. Precisely, we use an increasing sequence of times $(T_i)_{i \geq 1}$ such that the learning rule performs an independent EXP3.IX algorithm during each period $[T_i, T_{i+1})$. During this period, the EXP3.IX learner is run with i arms consisting of the experts E_k for $k \leq i$. Choosing $T_i = \sum_{j<i} j^3 = \frac{i^2(i+1)^2}{4}$ yields the following bounds.

Corollary 5.1. *There is an online learning rule* EXPINF *using bandit feedback such that for any countably infinite set of experts* $\{E_1, E_2, \ldots\}$ *(possibly randomized), for any* $T \geq 1$ *and* $0 < \delta \leq \frac{1}{2}$*, with probability at least* $1 - \delta$*,*

$$\max_{1 \leq i \leq T^{1/8}} \sum_{t=1}^{T} (r_t(E_{i,t}) - r_t(\hat{a}_t)) \leq cT^{3/4}\sqrt{\ln T} \ln \frac{T}{\delta}.$$

where $c > 0$ *is a universal constant. Further, with probability one on the learning and the experts, there exists* $\hat{T}$ *such that for any* $T \geq 1$*,*

$$\max_{1 \leq i \leq T^{1/8}} \sum_{t=1}^{T} (r_t(E_{i,t}) - r_t(\hat{a}_t)) \leq \hat{T} + cT^{3/4}(\ln T)^{3/2}.$$

Proof Denote by $(T_i = \sum_{j<i} j^3)_{i \geq 1}$ the restarting times used in the definition of EXPINF, and by $\hat{a}_t$ its selected action at time t. Theorem 5.6 implies that for any $i \geq 1$, with probability at least $0 < \delta < \frac{1}{2}$,

$$\max_{1 \leq j \leq i} \sum_{t=T_i}^{T_{i+1}-1} r_t(E_{j,t}) - r_t(\hat{a}_t) \leq c\sqrt{i(T_{i+1} - T_i) \ln i} \ln \frac{1}{\delta} = ci^2\sqrt{\ln i} \ln \frac{1}{\delta}.$$

Now fix $T \geq 1$ and $\delta > 0$. Let $i \geq 0$ such that $T_{i+1} \leq T < T_{i+2}$. Then summing the above equations gives that with probability at least δ,

$$\begin{aligned}\max_{1 \leq j \leq T^{1/8}} \sum_{t=1}^{T} r_t(E_{j,t}) - r_t(\hat{a}_t) &\leq T_{\lceil T^{1/8} \rceil} + (T - T_{i+1}) + \sum_{t=T_{\lceil T^{1/8} \rceil}}^{T_{i+1}-1} r_t(E_{i,t}) - r_t(\hat{a}_t) \\ &\leq T_{\lceil T^{1/8} \rceil} + (i+1) + c\frac{i(i+1)(2i+1)}{6}\sqrt{\ln i} \ln \frac{i}{\delta}.\end{aligned}$$

Now note that $i \sim \sqrt{2}T^{1/4}$ and $T_{\lceil T^{1/8} \rceil} \sim \frac{\sqrt{T}}{4}$ as $T \to \infty$. Therefore, there exists a universal constant $\tilde{c}$ such that for all $T \geq 1$, the right-hand term is upper bounded by $\tilde{c}T^{3/4}\sqrt{\ln T}\ln\frac{T}{\delta}$. This ends the proof of the first claim.

Now for any $T \geq 1$, using the probabilities of error $\delta_T = \frac{1}{T^2}$ which are summable, the Borel-Cantelli lemma implies that on an event of probability one, there exists $\hat{T}$ such that for any $T \geq \hat{T}$,

$$\max_{1 \leq j \leq T^{1/8}} \sum_{t=1}^{T} r_t(E_{j,t}) - r_t(\hat{a}_t) \leq \tilde{c}T^{3/4}\sqrt{\ln T}\ln(T^3) = 3\tilde{c}T^{3/4}\sqrt{\ln T}\ln T,$$

which ends the proof of the second claim by redefining the constant $c > 0$. ■

We briefly describe the implications of the algorithm EXPINF for learning under CS processes. A result from [Han21a, Lemma 24] showed that under CS processes there exists a sequence of policies that are dense (under $\mathbb{E}\hat{\mu}_{\mathbb{X}}$) within all measurable policies. Moreover, due to the relation between SMV and CS established in Proposition 5.1 above, we can directly infer this fact for the processes $\mathbb{X}^{(\leq M)}$ for $\mathbb{X} \in \text{SMV}$ (for any finite M). This is summarized in the following lemma.

Lemma 5.2 ([Han21a] Lemma 24)**.** *Let $\mathcal{X}$ be a separable metrizable Borel space and $\mathcal{A}$ a countable action space. There exists a sequence $\Pi = (\pi^l)_{l \geq 1}$ of measurable policies $\pi^l : \mathcal{X} \to \mathcal{A}$ such that for every $\mathbb{X} \in CS$ and measurable policy $\pi^\star : \mathcal{X} \to \mathcal{A}$,*

$$\inf_{l \geq 1} \mathbb{E}\big[\hat{\mu}_{\mathbb{X}}\big(\big\{x : \pi^l(x) \neq \pi^\star(x)\big\}\big)\big] = \inf_{l \geq 1} \mathbb{E}\left[\limsup_{T \to \infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\pi^l(X_t) \neq \pi^\star(X_t)]\right] = 0.$$

Moreover, for every $\mathbb{X} \in SMV$ and finite $M \geq 1$,

$$\inf_{l \geq 1} \mathbb{E}\big[\hat{\mu}_{\mathbb{X}^{(\leq M)}}\big(\big\{x \in \mathcal{X} : \pi^l(x) \neq \pi^\star(x)\big\}\big)\big] = 0.$$

As a result, under CS processes, one can restrict to a countable set of policies Π instead of all measurable policies. Plugging in this set of policies as the set of experts in EXPINF yields a universally consistent learning rule for CS processes (and naturally leads to a strategy for achieving sublinear regret on the $\mathcal{T}^{\leq M}$ times in $\mathbb{X}^{(\leq M)}$ for $\mathbb{X} \in \text{SMV}$). A full proof is given in Section 5.5.

5.4 Finite Action Spaces

In this section, we assume that the action space $\mathcal{A}$ is finite and we show that in this case, the set of processes $\mathbb{X}$ admitting universal learning is exactly SMV. In other words, we can recover the same processes that admit universal learning in the full-feedback setting. We start by showing that the SMV condition is necessary for universal consistency, which is a direct consequence of its necessity in the full-feedback case [Han21a].

Theorem 5.7. *If $2 \leq |\mathcal{A}| < \infty$, $\mathbb{X} \in SMV$ is necessary for universal consistency: SOCB $\subset$ SMV.*

Proof In the full-information feedback setting, [Han21a, Theorem 37] showed that $\mathbb{X} \in \text{SMV}$ is necessary for universal learning even for noiseless responses in binary classification. We will present a simple reduction from the full-feedback to the partial-feedback setting. Precisely, let $a_0, a_1 \in \mathcal{A}$ be two distinct actions. To any measurable function $f : \mathcal{X} \to \{0,1\}$ we associate a deterministic reward function $r_f : \mathcal{X} \times \mathcal{A} \to [0,1]$ as follows

$$r_f(x,a) = f(x)\mathbb{1}[a = a_1] + (1 - f(x))\mathbb{1}[a = a_0], \quad x \in \mathcal{X}, a \in \mathcal{A}.$$

Note that any action $a \in \mathcal{A} \setminus \{a_0, a_1\}$ always has reward 0. Now suppose that for a process $\mathbb{X}$ there exists a universally consistent learning rule $f.$ for contextual bandits. Then, we can consider the following learning rule for the complete-feedback setting, recursively defined as

$$\tilde{f}_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{y}_{\leq t-1}, x_t) = \mathbb{1}[f_t(\boldsymbol{x}_{\leq t-1}, (\mathbb{1}[\tilde{f}_i(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}, x_i) = y_i])_{i \leq t-1}, x_t) = a_1].$$

for any $t \geq 1$, $\boldsymbol{x}_{\leq t} \in \mathcal{X}^{t-1}$ and $\boldsymbol{y}_{\leq t-1} \in \{0,1\}^{t-1}$. We now show that $\tilde{f}.$ is universally consistent for the noiseless full-feedback setting. For any measurable function $f : \mathcal{X} \to \{0,1\}$, the learning rule $f.$ is consistent for the rewards r_f. In particular, if we denote by $\hat{a}_t$ the action selected by $f.$ at time t, using the measurable policy $\pi_f : x \in \mathcal{X} \mapsto a_0\mathbb{1}[f(x) = 0] + a_1\mathbb{1}[f(x) = 1] \in \mathcal{A}$ which always selects the best action we obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi_f(X_t)) - r_t(\hat{a}_t) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\hat{a}_t \neq \pi_f(X_t)] \leq 0, \quad (a.s.).$$

Now consider the actions $\hat{a}_t$ selected under $\mathbb{X}$ and rewards r_f and denote by $\tilde{Y}_t$ the prediction of $\tilde{f}.$ at time t under $\mathbb{X}$ and values $Y_t = f(X_t)$ for $t \geq 1$. By construction, for any $t \geq 1$, we have $\mathbb{1}[\hat{a}_t \neq \pi_f(X_t)] \geq \mathbb{1}[\tilde{Y}_t \neq f(X_t)]$. Then, almost surely $\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\tilde{Y}_t \neq f(X_t)] \underset{n\to\infty}{\longrightarrow} 0$. This shows that $\tilde{f}.$ is universally consistent for noiseless responses in binary classification, hence $\mathbb{X} \in \text{SMV}$, which completes the proof. ■

Before providing our optimistically universally consistent learning rule, we provide some intuition on the algorithmic challenges and a brief overview of our algorithm structure.

Limitations of generalization-based strategies. As mentioned in Section 5.3.2, for CS processes, one can use a traditional generalization-based approach via the EXPINF algorithm with a dense set of policies, to achieve universal consistency. This strategy, in which one selects a set of policies Π and ensures low regret for an increasing number of policies within Π is insufficient for SMV processes. First, observe that Π must be countable. Further, if $M(T)$ is the number of policies considered at time T, one should have $\log M(T) = o(T)$ to ensure sublinear regret, because the regret of standard expert bandit strategies (e.g. EXP4) for T steps typically scales as $\sqrt{T \log M(T)}$. Unfortunately, it can be shown that the processes that admit a countable empirically dense set Π of policies as per Lemma 5.2 are exactly CS processes. Further, the constraint $\log M(T) = o(T)$ is also prohibitive. For instance one can design a deterministic SMV process as follows: let $(x_k)_{k\geq 1}$ a sequence of distinct points in $\mathcal{X}$, and $(d_k)_{k\geq 1}$ a non-decreasing sequence of integers to be chosen later. We consider the process $\mathbb{X}$ that successively visits x_k duplicated d_k times. The process belongs to SMV whenever

the number of duplicates diverges, which ensures that the number of visited distinct points $k(T)$ after T steps is sublinear. The sublinear rate of $k(T)$ can, however, be arbitrarily slow. For a sufficiently slow diverging sequence $(d_k)_{k\geq 1}$, one can have $\log M(T) = o(k(T))$. Then, the set of considered policies is not large enough to contain a significant fraction of relevant policies—at the high level, all functions $\{x_k, k \leq k(T)\} \to \mathcal{A}$ are relevant policies. As a result, this approach cannot be universally consistent even for all deterministic SMV processes.

Limitations of personalization-based strategies. A natural strategy for deterministic SMV processes is pure personalization, that is assigning an independent bandit learner to each distinct context observed. This strategy is only consistent if the number of duplicates generally diverges so that the average regret incurred by each independent bandit learner decays to 0. Unfortunately, this is of course not the case for all SMV processes, for instance, i.i.d. processes may never visit the same point twice. For all these processes, one cannot disregard the information provided by neighboring contexts and needs a more population-level approach.

Overview of the optimistically universal algorithm structure. To overcome the challenges of naïve approaches, we aim to balance both strategies mentioned above. The strategy heavily relies on Proposition 5.1, which states that for SMV processes $\mathbb{X}$, if one fixes a maximum cap M for the number of duplicates, the resulting sparsified process $\mathbb{X}^{(\leq M)}$ is CS. As a result, one can safely use the generalization-based approach for the $\mathcal{T}^{\leq M}$ times in the duplicate-capped processes $\mathbb{X}^{(\leq M)}$. Because the number of duplicates can be arbitrarily large, one needs to consider these processes for arbitrarily large values of M. For reasons to be detailed later, we consider values M from the sequence $\{4^p, p \geq 0\}$. Instead of considering overlapping times $\mathcal{T}^{\leq 4^p}$, we instead introduce disjoint families of times $\mathcal{T}^{<4^{p+1}} \setminus \mathcal{T}^{<4^p}$, i.e., times corresponding to duplicates of index falling in the interval $[4^p, 4^{p+1})$. These will be called times of category p, and the corresponding sparsified process denoted as $\mathbb{X}^{(p)} = \mathbb{X}^{\mathcal{T}^{<4^{p+1}} \setminus \mathcal{T}^{<4^p}}$.

As a result of this procedure, we decompose a process $\mathbb{X} \in \text{SMV}$ into a countable set of sparsified processes $(\mathbb{X}^{(p)})_{p\geq 0}$, one for each category of times. The key observation is that while the generalization strategy (strategy 1) would achieve sublinear regret on the $\mathcal{T}^{<4^{p+1}} \setminus \mathcal{T}^{<4^p}$ times of each one of them $\mathbb{X}^{(p)}$ because of their CS property, a learner does not know in advance their rate of convergence. The individual convergence for $\mathbb{X}^{(p)}$ can be arbitrarily slow, and because there is an infinite set of such sparsified processes for $p \geq 0$, using the generalization-based strategy for all of them may not be consistent. On the other hand, for high values of p, one has access to many duplicates, hence, using a pure personalization approach (strategy 0) on $\mathbb{X}^{(p)}$ yields a low average regret. This average regret is guaranteed by classical bandit guarantees (at least in expectation), hence is "safe". However, it is non-negligible: the bandit learner has access to fewer than 4^{p+1} duplicates, hence one should expect an average regret of at least $1/2^{p+1}$. Eventually, one should therefore use strategy 1 to be consistent. Fortunately, because of the choice of sequences for M, the safe average regret of strategy 0 decays sufficiently quickly as p increases. This allows for the following overall strategy for the process $\mathbb{X}^{(p)}$:

Always: Estimate the performance of strategy 1 compared to strategy 0. This is done by using sparse exploration times, designed solely to estimate the rewards obtained by each strategy.

Step 1: Use strategy 0 by default, which safely ensures relatively low average regret—at most $1/c^p$ for some universal constant $c > 1$.

Step 2: Whenever strategy 1 is estimated to have similar performance as strategy 0, we switch to strategy 1.

Step 3: If strategy 1 shows worse estimated performance than strategy 0 at some point, we switch to strategy 0 for an extended period and then go back to Step 1.

We briefly mention some implementation difficulties. First, we aim to estimate the performance of *strategies*. As opposed to experts, these strategies are adaptive algorithms, hence standard importance sampling techniques are not sufficient to yield adequate estimators. To estimate strategy 0, which assigns bandit learners to each context, we instead need to randomly assign a context and all its duplicates for the estimation of the reward of strategy 0, and use the reward of a bandit learner on these duplicates to estimate the performance of the complete strategy 0. As a result, we always ensure that duplicates from the same context are assigned the same purpose: times dedicated to estimating the performance of either strategy 0 or strategy 1, or non-exploration times. We note that this is not necessary for estimating the performance of strategy 1 since it is essentially a combination of experts—it is sufficient to estimate the performance of each hypothesis policy in strategy 1. However, this is also necessary to ensure that one does not affect the performance of the algorithm on non-exploration times when strategy 0 was selected.

Second, we give some intuition for Step 3. We recall that the performance of strategy 1 is "uncertain" in that one does not know in advance when its excess average regret converges to 0. In particular, although strategy 1 is estimated to have better performance at a given period, this may not be the case in future times. The algorithm needs to detect such failures in performance to not incur significant excess loss, and quickly switch to strategy 0 again. To compensate for the mistake when the algorithm used strategy 1 when strategy 0 had a higher reward, we ensure that the algorithm then uses the safe strategy 0 for a sufficiently "long time", before considering switching to strategy 1 again.

Last, the estimation and implementation of these strategies require adequate scheduling. To do so, we partition times into periods, estimate the performance during each period, and select the strategy 0 or 1 for the next period. Because our objective is the average regret, the natural scale is exponential. Roughly speaking, we partition the space according to sequences of the form $(1 + \alpha_p)^q{}_{q \geq 1}$ so that each period similarly affects the average regret objective—roughly between $-\alpha_p$ and α_p. Conversely, there is little advantage in using finer partitions (e.g. polynomial), since the average reward of the strategies can only be non-trivially modified through periods of such exponential sizes. As p grows, these periods are more refined so that the estimation process is faster, which is needed for Step 3: roughly speaking, we decrease the term α_p as p grows.

Detailed exposition of the algorithm. We now formally present our learning rule for contextual bandits, which we will next show is universally consistent under any SMV process. This learning rule has different behavior depending on the number of past duplicates for each context. Precisely, for any time t, we compute a corresponding category p such that the number of past occurrences of X_t belongs in the interval $[4^p, 4^{p+1})$. The learning rule will treat times from different categories completely separately. The formal definition is given by the function below

$$\textsc{Category}(t, \mathbb{X}_{\leq t}) = \left\lfloor \log_4 \left(\sum_{t' \leq t} \mathbb{1}[X_{t'} = X_t] \right) \right\rfloor .$$

For convenience we may write $\textsc{Category}(t)$ instead of $\textsc{Category}(t, \mathbb{X}_{\leq t})$. Further, for a given category p, the algorithm will proceed by periods $[T_p^q, T_p^{q+1})$ defined as follows. For any $p \geq 0$ and $q \geq p2^p$, we define the times $T_p^q = 2^k + \frac{i}{2^p} 2^k$, where $q = k2^p + i$ with $0 \leq i < 2^p$. Note that the sequence $(T_p^q)_q$ has an exponential behaviour with rate between 2^{-p-1} and 2^{-p}. We will refer to $[T_p^q, T_p^{q+1})$ as the period q for category p. Let $\textsc{Period}(t)$ be the function that returns the index q such that $T_p^q \leq t < T_p^{q+1}$ where p is the category of t. An illustration of these category and period constructions is given in Fig. 5.1.

Now let $(\pi^l)_{l \geq 1}$ be a sequence of measurable functions from $\mathcal{X}$ to $\mathcal{A}$ that are dense within measurable functions under CS processes, as given by Lemma 5.2. Intuitively, the learning rule combines two strategies: strategy 0 which applies a separate EXP3 algorithm to each distinct instance, and strategy 1 which performs the best policy within a subset of the policies $(\pi^l)_{l \geq 1}$. To know which strategy to apply, the learning rule estimates the counterfactual loss of strategy i, using classical importance sampling on some allocated exploration times for strategy i. In exploitation times, the learning rule uses these estimates to perform the best strategy.

We first define the procedure $\textsc{AssignPurpose}$ which takes as input a time t and determines whether this time will be used for exploration of strategy 0 (output 0), strategy 1 (output 1), or exploitation (output 2). Intuitively, $\textsc{AssignPurpose}$ selects exploration times randomly with small probability while ensuring that times t, t' from the same category p, period q, and that are duplicates $X_t = X_{t'}$ are assigned the same output, hence will serve for the same exploration or exploitation purpose. The algorithm is formally defined in Algorithm 5.1.

Next, we define the subroutine $\textsc{Explore}(i; t)$ that will be called on exploration times t for strategy i. We start with $i = 0$. The subroutine updates an estimator $\hat{R}_p^0(q)$ of the loss that would be incurred by using strategy 0 for all times in category p during period q. $\textsc{Explore}(0, \cdot)$ is defined formally in Algorithm 5.2.

Then, we define $\textsc{Explore}(1, \cdot)$. It updates an estimator $\hat{R}_p^l(q)$ of the loss that would have been incurred using the policy π^l for all times in category p during period q, for all $l \geq 1$. Because there is an infinite number of such policies, they are introduced sequentially in the estimation process. $\textsc{Explore}$ is defined formally in Algorithm 5.3.

The estimates $\hat{R}_p^l(q)$ updated by $\textsc{Explore}$ are then used to select the strategy to perform on exploitation times. For any category $p \geq 0$, before starting phase q, the learning rule commits to performing strategy $\mathcal{P}_p(q) \in \{0, 1\}$, for times of that phase q for category p. The choice of strategy $\mathcal{P}_p(q)$ is performed by a subroutine $\textsc{SelectStrategy}$ which applies an

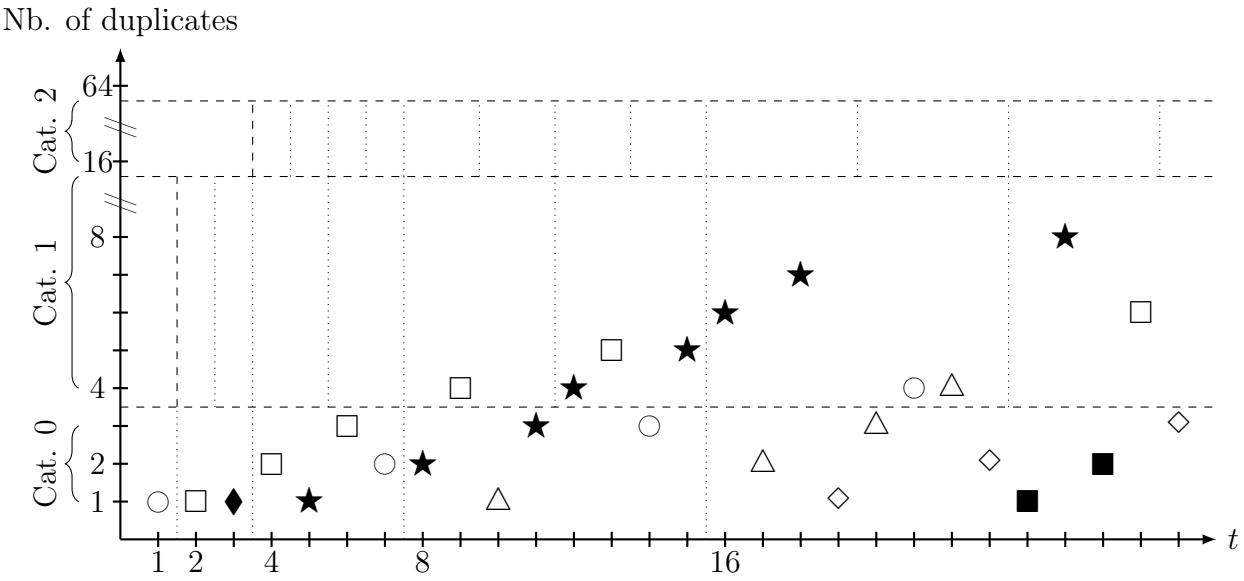

Figure 5.1: Illustration of the functions CATEGORY and PERIOD. The plot represents a sequence of contexts, where different contexts are represented by different markers (○, □, ⧫, ★, △, ◇, and ■). Times in category p have a total number of past duplicates falling in $[4^p, 4^{p+1})$, as represented by the horizontal dashed lines. For each category p, times are grouped along periods, represented by the vertical dotted lines. These periods follow an exponential scale $(T_p^q)_{q\geq 1}$ that is refined as the category p increases. For $p = 0$, the periods start at powers of 2, while periods for category p are twice more refined than periods from category $p+1$. For convenience, we represented periods $[T_p^q, T_p^{q+1})$ only starting from $q \geq 2^p$ as shown by the vertical dashed lines—for $q < 2^p$, these periods are not integral and would not contain any times anyway.

Input: time t, $\mathbb{X}_{\leq t}$, CATEGORY(t') for $t' \leq t$, ASSIGNPURPOSE(t') for $t' < t$.
Output: ASSIGNPURPOSE$(t) \in \{0, 1, 2\}$.
$p =$ CATEGORY(t); $q =$ PERIOD(t)
if *exists* $t' < t$ *with* CATEGORY$(t') = p$; PERIOD$(t') = q$ *and* $X_t = X_{t'}$ **then** `// Not the first occurrence of` X_t `in current period`
  Return ASSIGNPURPOSE(t')
else `// First occurrence of` X_t `in current period`
  $p_t = 1/(2t^{1/4})$, $U_t \sim \mathcal{U}([0, 1])$
  if $U_t \leq p_t$ **then** Return 0 `// Exploration for strategy 0;`
  else if $p_t < U_t \leq 2p_t$ **then** Return 1 `// Exploration for strategy 1;`
  else Return 2 `// Exploitation;`
end

Algorithm 5.1: ASSIGNPURPOSE

$\eta_p = \mathcal{O}(2^{-p/2})$ average reward penalty for strategy 0 then select the strategy that obtained the highest adjusted estimated reward during the previous period. This penalty is used to favor strategy 1 since it should eventually be used instead of strategy 0, as discussed in the

Input: time t, $\mathbb{X}_{\leq t}$, $\textsc{Category}(t')$ for $t' \leq t$, rewards $\boldsymbol{r}_{<t}$, $\hat{R}^0_p(q)$ for $p \geq 0, q \geq p2^p$.
Output: Selects action $\hat{a}_t$ and updates $\hat{R}^0_p(q)$ for $p = \textsc{Category}(t)$, $q = \textsc{Period}(t)$.
$p = \textsc{Category}(t)$, $q = \textsc{Period}(t)$
$S_t = \{t' < t : \textsc{Category}(t') = p, \textsc{Period}(t') = q, X_{t'} = X_t\}$
$\hat{a}_t = \text{EXP3}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$
Receive reward r_t
Let $t' = \min S_t$ `// First occurrence of` X_t
$\hat{R}^0_p(q) \leftarrow \hat{R}^0_p(q) + \frac{r_t}{p_{t'}}$ `// Update estimate` $\hat{R}^0_p(q)$

Algorithm 5.2: $\textsc{Explore}(0; \cdot)$

Input: time t, $\mathbb{X}_{\leq t}$, $\textsc{Category}(t')$ for $t' \leq t$, rewards $\boldsymbol{r}_{<t}$, $\hat{R}^l_p(q)$ for $l \geq 1, p \geq 0, q \geq p2^p$.
Output: Selects action $\hat{a}_t$ and updates $\hat{R}^l_p(q)$ for $p = \textsc{Category}(t)$, $q = \textsc{Period}(t)$.
$p = \textsc{Category}(t)$, $q = \textsc{Period}(t)$, $k = \lfloor \log_2 t \rfloor$
$l_t = \mathcal{U}(\{1, \ldots, k\})$ `// Uniform exploration`
$\hat{a}_t = \pi^{l_t}(X_t)$
Receive reward r_t
$t' = \min\{s < t : \textsc{Category}(s) = p, \textsc{Period}(s) = q, X_s = X_t\}$ `// First occurrence of` X_t
$\hat{R}^l_p(q) \leftarrow \hat{R}^l_p(q) + \frac{k}{p_{t'}} r_t \mathbb{1}[l = l_t], \quad 1 \leq l \leq k$ `// Update estimate` $\hat{R}^{l_t}_p(q)$

Algorithm 5.3: $\textsc{Explore}(1; \cdot)$

overview of the algorithm. Last, if during the current period q, strategy 0 obtained the highest adjusted reward, we select this strategy for the future periods $q < q' \leq q + p2^p$. This ensures that if by mistake the rule selected $\mathcal{P}_p(q) = 1$, the loss incurred during this period is mitigated for the next strategy selection: the current performance until time T^{q+1} is negligible up to a small average loss starting from time $T_p^{q+2^p+1}$. The construction of $\textsc{SelectStrategy}$ is detailed in Algorithm 5.4.

We are now ready to define the learning rule for stochastic rewards. On exploration times, the learning rule calls the subroutine $\textsc{Explore}$, and on exploitation times, the learning rule performs the corresponding strategy $\mathcal{P}_p(q)$ for times in category p during phase q. The construction of the learning rule is detailed in Algorithm 5.5.

The main result of this section is that this learning rule is optimistically universal.

Theorem 5.8. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathcal{A}$ be a finite action set. There exists an optimistically universal learning rule (Algorithm 5.5) and the learnable processes are $\textsc{Socb} = \textsc{SMV}$.*

Proof We will denote by $\hat{a}_t$ the action selected by the learning rule at time t. For any $p \geq 0$, we define the set $\mathcal{T}_p$ of times in category p as follows

$$\mathcal{T}_p = \left\{ t \geq 1 : 4^p \leq \sum_{t' \leq t} \mathbb{1}[X_{t'} = X_t] < 4^{p+1} \right\},$$

Input: Category p, phase q, variable states $\hat{R}_p^l(t)$ for $t < T_p^{q+1}$
Output: Selects strategy $\mathcal{P}_p(r)$ for some future phases $r > q$.
$\eta_p = 10\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}$, $k = \lfloor \log_2 T_p^q \rfloor$
if $\mathcal{P}_p(q+1)$ *has not been defined yet* **then**
  if $\hat{R}_p^0(q) - \eta_p(T_p^{q+1} - T_p^q) \geq \max_{1\leq l\leq k} \hat{R}_p^l(q)$ **then**
    $\mathcal{P}_p(q') = 0, \quad q < q' \leq q + p2^p$ `// perform strategy 0 until current performance is negligible up to a` $\mathcal{O}(2^{-p})$ `average loss`
  else
    $\mathcal{P}_p(q+1) = 1$
  end
end

Algorithm 5.4: SELECTSTRATEGY

i.e. the set of times which correspond to duplicates with index in $[4^p, 4^{p+1})$. We also define

$$\mathcal{T}_p^{exp,i} = \{t \geq 2^{32p} : \textsc{AssignPurpose}(t) = i\}, \quad i \in \{0,1\},$$
$$\tilde{\mathcal{T}}_p = \{t \geq 2^{32p} : \textsc{AssignPurpose}(t) = 2\},$$

the set of exploration times for strategy i in category p, and exploitation times in category p, respectively. For convenience, we also define $\mathcal{T}_p(q) = \mathcal{T}_p \cap [T_p^q, T_p^{q+1})$ times in category p and phase q. Last, we define $A_p(q) = |\mathcal{T}_p(q) \cap (\mathcal{T}_p^{exp,0} \cup \mathcal{T}_p^{exp,1})|$ the number of exploration times in period q for category p.

Now fix a process $\mathbb{X} \in \text{SMV}$ and let r be a reward mechanism on $\mathcal{A}\times\mathcal{X}$. We recall the notation $\bar{r}(\cdot,\cdot) = \mathbb{E}[r(\cdot,\cdot)]$ for the average reward. We aim to show that $f.$ is consistent under $\mathbb{X}$ for the rewards given by r. We first define the policy π^* given by $\pi^*(x) = \arg\max_{a\in\mathcal{A}} \bar{r}(a,x)$, where ties are broken by the lexicographic rule. This function is measurable given that $\mathcal{A}$ is finite. Further, it is an optimal policy in the sense that for any measurable function $\pi : \mathcal{X} \to \mathcal{A}$ and any $x \in \mathcal{X}$, $\bar{r}(\pi(x), x) \leq \bar{r}(\pi^*(x), x)$.

For $p \geq 0$, we first analyze the reward estimates $\hat{R}_p^l(q)$ for $q \geq p2^{p+5}$ ($T_p^{p2^{p+5}} = 2^{32p}$) and $l \geq 0$. First note that the exploration times $\mathcal{T}_p^{exp,0}$ and $\mathcal{T}_p^{exp,1}$ were constructed precisely so that times corresponding to the same instance and within the same period, fall in the same set $\mathcal{T}_p^{exp,0}$, $\mathcal{T}_p^{exp,1}$, or $\tilde{\mathcal{T}}_p$. For simplicity, we will write $\mathcal{X}_p(q) = \{X_t, t \in \mathcal{T}_p(q)\}$ the set of visited instances during period q of category p, and for $x \in \mathcal{X}_p(q)$ we denote $t_p(q;x) = \min\{t \in \mathcal{T}_p(q) : X_t = x\}$ the first time of occurrence of x in period q. Then, we can write

$$\hat{R}_p^0(q) = \sum_{x\in\mathcal{X}_p(q)} \frac{\mathbb{1}[U_{t_p(q;x)} \leq p_{t_p(q;x)}]}{p_{t_p(q;x)}} \sum_{t\in\mathcal{T}_p(q), X_t = x} \tilde{r}_t$$

where $\tilde{r}_t$ is the reward at time t that would have been obtained by performing strategy 0 during period q, i.e., assigning an independent EXP3 learner for each different instance in this period. We compare $\hat{R}_p^0(T)$ to the average reward obtained by the optimal policy π^*,

$$\bar{R}_p^*(q) := \sum_{t\in\mathcal{T}_p(q)} \bar{r}(\pi^*(X_t), X_t).$$

$\hat{R}_p^l = 0, l \geq 0, p \geq 0$; $\mathcal{P}_p(p2^{p+5}) = 0, p \geq 0$ `// Initialization`
for $t \geq 1$ **do**
  Observe context X_t
  $p = \text{CATEGORY}(t)$, $q = \text{PERIOD}(t)$
  if $t < 2^{32p}$ **then** `// Initially perform strategy 0 without period restriction`
    $S_t = \{t' < t : \text{CATEGORY}(t') = p, X_{t'} = X_t\}$
    $\hat{a}_t = \text{EXP3}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$
  else if $i := \textit{ASSIGNPURPOSE}(t) \leq 1$ **then**
    $\text{EXPLORE}(i; t)$
  else `// Perform strategy` $\mathcal{P}_p(q)$
    if $\mathcal{P}_p(q) = 0$ **then**
      $S_t = \{t' < t : \text{CATEGORY}(t') = p, \text{PERIOD}(t') = q, X_{t'} = X_t\}$
      $\hat{a}_t = \text{EXP3}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$
    else
      $k = \lfloor \log_2 T_p^q \rfloor$
      $S_t = \{t' < t : \text{CATEGORY}(t') = p, \text{PERIOD}(t') = q, \text{ASSIGNPURPOSE}(t') = 2\}$
      $l_t = \text{EXP3.IX}_{\{1,\ldots,k\}}(\boldsymbol{l}_{S_t}, \boldsymbol{r}_{S_t})$ `// Select policy` π^{l_t}
      $\hat{a}_t = \pi^{l_t}(X_t)$
    end
    Receive reward r_t
  end
  $\mathcal{E} = \{(p', q') : q' \geq p' 2^{p'+5}, t = T_{p'}^{q'+1} - 1\}$
  for $(p', q') \in \mathcal{E}$ **do**
    $\text{SELECTSTRATEGY}(p', q')$ `// At the end of a phase` $[T_{p'}^{q'}, T_{p'}^{q'-1})$, `select strategy for future phases`
  end
end

Algorithm 5.5: An optimistically universal learning rule for stochastic rewards

Observe that conditionally on $\mathbb{X}$, the terms in the sum of $\hat{R}_p^0(q)$ are independent. For any $x \in \mathcal{X}_p(q)$, let $\bar{R}_p^0(q; x) = \mathbb{E}[\sum_{t \in \mathcal{T}_p(q), X_t = x} \tilde{r}_t \mid \mathbb{X}]$, the average reward obtained by strategy 0 on the instance x. We will use the notation $N_p(q; x) = |\{t \in \mathcal{T}_p(q), X_t = x\}| \leq 4^{p+1}$ for the number of occurrences of the instance x within $\mathcal{T}_p$. Note that

$$\left| \frac{\mathbb{1}[U_{t_p(q;x)} \leq p_{t_p(q;x)}]}{p_{t_p(q;x)}} \sum_{t \in \mathcal{T}_p(q), X_t = x} \tilde{r}_t \right| \leq \frac{N_p(q; x)}{p_{t_p(q;x)}} \leq 2^{2p+3} (T_p^{q+1})^{1/4},$$

and that $|\mathcal{X}_p(q)| \leq \frac{T_p^{q+1}}{2^{2p}}$ since by definition of $\mathcal{T}_p$ each instance has already occurred 4^p times. As a result, we can apply Hoeffding's inequality to obtain

$$\mathbb{P}\left[\left| \hat{R}_p^0(q) - \sum_{x \in \mathcal{X}_p(q)} \bar{R}_p^0(q; x) \right| \leq (T_p^{q+1})^{\frac{7}{8}} \mid \mathbb{X} \right] \geq 1 - 2\exp\left(-\frac{(T_p^{q+1})^{1/4}}{2^{2p+5}} \right) := 1 - 2p_1(p, q)$$

Now applying Theorem 5.5 to each pseudo-regret $\bar{R}_p^0(q;x)$ yields

$$\begin{aligned}
&\sum_{x\in\mathcal{X}_p(q)} \bar{R}_p^0(q;x) \\
&\geq \bar{R}_p^*(q) - 2\sqrt{\frac{|\mathcal{A}|\ln|\mathcal{A}|}{2^{p/2}}}(T_p^{q+1}-T_p^q) - 2\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\sum_{x\in\mathcal{X}_p(q),N_p(q;x)\leq 2^{p/2}} N_p(q;x) \\
&\geq \bar{R}_p^*(q) - 2\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1}-T_p^q) - 2\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\frac{2^{p/2}}{4^p}T_p^{q+1} \\
&\geq \bar{R}_p^*(q) - 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1}-T_p^q).
\end{aligned}$$

where in the third inequality, we used the fact that instances appearing in $\mathcal{T}_p$ before T_p^{q+1} are visited at least 4^p times before horizon T_p^{q+1}, by construction of $\mathcal{T}_p$; and in the last inequality we used $2^{-p-1}T_p^{q+1} \leq T_p^{q+1} - T_p^q \leq 2^{-p}T_p^q$. Also, note that $\bar{R}_p^*(q) \geq \sum_{x\in\mathcal{X}_p(q)} \bar{R}_p^0(q;x)$. As a result, taking the expectation over $\mathbb{X}$, we obtain that with probability at least $1-2p_1(p,q)$,

$$\left|\hat{R}_p^0(q) - \bar{R}_p^*(q)\right| \leq (T_p^{q+1})^{\frac{7}{8}} + 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1}-T_p^q). \tag{5.1}$$

Now consider the quantity $\tilde{R}_p^0(q)$, the reward that would be obtained for exploitation times on period q if strategy 0 was applied. We have

$$\tilde{R}_p^0(q) = \sum_{x\in\mathcal{X}_p(q)}\sum_{t\in\mathcal{T}_p(q),X_t=x,t\in\tilde{\mathcal{T}}_p} \tilde{r}_t \geq \sum_{x\in\mathcal{X}_p(q)}\sum_{t\in\mathcal{T}_p(q),X_t=x} \tilde{r}_t - A_p(q).$$

Similarly as above, using Hoeffding's inequality, we have

$$\mathbb{P}\left[\sum_{x\in\mathcal{X}_p(q)}\sum_{t\in\mathcal{T}_p(q),X_t=x} \tilde{r}_t \geq \sum_{x\in\mathcal{X}_p(q)} \bar{R}_p^0(q;x) - (T_p^{q+1})^{3/4}\right] \geq 1 - e^{-\frac{\sqrt{T_p^{q+1}}}{2^{2p+3}}} := 1-p_2(p,q).$$

As a result, with probability $1-p_2(p,q)$, we have

$$\tilde{R}_p^0(q) \geq \bar{R}_p^*(q) - 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1}-T_p^q) - (T_p^{q+1})^{3/4} - A_p(q). \tag{5.2}$$

We now turn to the estimates $\hat{R}_p^l(q)$ for $l \geq 1$. Note that the estimation of $R_p^l(q)$ only starts at time 2^l. Hence, we can consider $k(q) = \lfloor \log_2 T_p^q \rfloor = \lfloor \frac{q}{2^p} \rfloor$ and observe that during period q, the only estimates $\hat{R}_p^l(q)$ that are considered are for $1 \leq l \leq k(q)$. Therefore, similarly as for the estimates $\hat{R}_p^0(q)$, we can write for $q \geq p2^{p+5}$ and $1 \leq l \leq k(q)$,

$$\hat{R}_p^l(q) = \sum_{x\in\mathcal{X}_p(q)} \frac{\mathbb{1}[p_{t_p(q;x)} < U_{t_p(q;x)} \leq 2p_{t_p(q;x)}]}{p_{t_p(q;x)}} \sum_{t\in\mathcal{T}_p(q)X_t=x} k(t)\mathbb{1}[l=l_t]r(\pi^l(x),x),$$

where $k(t)$ is the number of policies π^l tested at time t, i.e. $k(t) = \lfloor \log_2 t \rfloor$. Conditionally on $\mathbb{X}$ and $\boldsymbol{U}$ we can apply Hoeffding's inequality to obtain

$$\mathbb{P}\left[\left|\hat{R}_p^l(q) - \sum_{x\in\mathcal{X}_p(q)} \sum_{t\in\mathcal{T}_p(q),X_t=x} \frac{\mathbb{1}[p_{t_p(q;x)} < U_{t_p(q;x)} \le 2p_{t_p(q;x)}]}{p_{t_p(q;x)}} \bar{r}(\pi^l(x),x)\right| \le (T_p^{q+1})^{7/8} \mid \mathbb{X}, \boldsymbol{U}\right]$$
$$\ge 1 - 2e^{-\frac{2(T_p^{q+1})^{7/4}}{(T_p^{q+1}-T_p^q)4(\log_2 T_p^{q+1})^2\sqrt{T_p^{q+1}}}} \ge 1 - 2e^{-\frac{2^p(T_p^{q+1})^{1/4}}{4(\log_2 T_p^{q+1})^2}} := 1 - 2p_3(p,q).$$

For convenience, let us denote by $\hat{R}_{p,bis}^l(q)$ the sum in the above inequality. We also define $\bar{R}_p^l(q) = \sum_{t\in\mathcal{T}_p(q)} \bar{r}(\pi^l(X_t),X_t)$ the expected reward of policy l on period q. Now, as before, we have

$$0 \le \sum_{t\in\mathcal{T}_p(q),X_t=x} \frac{\mathbb{1}[p_{t_p(q;x)} < U_{t_p(q;x)} \le 2p_{t_p(q;x)}]}{p_{t_p(q;x)}} \bar{r}(\pi^l(x),x) \le \frac{N_p(q;x)}{p_{t_p(q;x)}} \le 2^{2p+3}(T_p^{q+1})^{1/4}.$$

As a result, conditionally on $\mathbb{X}$, Hoeffding's inequality yields

$$\mathbb{P}[|\hat{R}_{p,bis}^l(q) - \bar{R}_p^l(q)| \le (T_p^{q+1})^{7/8} \mid \mathbb{X}] \ge 1 - 2p_1(p,q).$$

Thus, with probability at least $1 - 2p_1(p,q) - 2p_3(p,q)$ we have

$$|\hat{R}_p^l(q) - \bar{R}_p^l(q)| \le (T_p^{q+1})^{7/8}. \tag{5.3}$$

Next, we consider the quantity $\tilde{R}_p^1(q)$, the reward that would have been obtained for exploitation times on period q if strategy 1 was applied. Then, using Theorem 5.6, we have with probability at least $1 - e^{-(T_p^{q+1})^{1/4}} := 1 - p_4(p,q)$,

$$\max_{1\le l\le k(q)} \sum_{t\in\mathcal{T}_p(q)\cap\tilde{\mathcal{T}}_p} r_t(\pi^l(X_t),X_t) - \tilde{R}_p^1(q)$$
$$\le c\sqrt{k(q)\ln k(q)(T_p^{q+1}-T_p^q)}(T_p^{q+1})^{1/4} \le c(T_p^{q+1})^{3/4}\ln T_p^{q+1}.$$

As a result, we have

$$\tilde{R}_p^1(q) \ge \max_{1\le l\le k(q)} \sum_{t\in\mathcal{T}_p(q)} r_t(\pi^l(X_t),X_t) - c(T_p^{q+1})^{3/4}\ln T_p^{q+1} - A_p(q).$$

By Hoeffding's inequality, for $1 \le l \le k(q)$, with probability at least $1 - e^{-2^p\sqrt{T_p^{q+1}}} := 1 - p_5(p,q)$,

$$\sum_{t\in\mathcal{T}_p(q)} r_t(\pi^l(X_t),X_t) \ge \bar{R}_p^l(q) - (T_p^{q+1})^{3/4}.$$

Hence, with probability $1 - p_4(p,q) - k(q)p_5(p,q)$ we have

$$\tilde{R}_p^1(q) \ge \max_{1\le l\le k(q)} \bar{R}_p^l(q) - (T_p^{q+1})^{3/4} - c(T_p^{q+1})^{3/4}\ln T_p^{q+1} - A_p(q). \tag{5.4}$$

We will also need the quantity $\tilde{R}_p^1(q;T)$ for $T_p^q \le T < T_p^{q+1}$ which is the reward that would have been obtained for exploitation times from T_p^q to T. The same arguments as above show that with probability at least $1 - p_4(p,q) - k(q)p_5(p,q)$ we have

$$\tilde{R}_p^1(q;T) \ge \max_{1\le l\le k(q)} \sum_{t\in\mathcal{T}_p(q), t\le T} \bar{r}(\pi^l(X_t), X_t) - (T_p^{q+1})^{3/4} - c(T_p^{q+1})^{3/4} \ln T_p^{q+1} - A_p(q). \quad (5.5)$$

Last, we now bound the exploration terms $A_p(q)$ to show that exploration times are negligible. Writing $A_p(q) = \sum_{x\in\mathcal{X}_p(q)} \mathbb{1}[U_{t_p(q;x)} \le 2p_{t_p(q;x)}] N_p(q;x)$, and because $\frac{N_p(q;x)}{t_p(q;x)^{1/4}} \le 2^{2p+2}(T_p^{q+1})^{1/4}$, using Hoeffding's inequality we obtain that with probability at least $1 - e^{-\frac{(T_p^{q+1})^{1/4}}{2^{2p+3}}} := 1 - p_6(p,q)$,

$$A_p(q) \le \sum_{x\in\mathcal{X}_p(q)} \frac{N_p(q;x)}{t_p(q;x)^{1/4}} + (T_p^{q+1})^{7/8} \le \frac{T_p^{q+1} - T_p^q}{(T_p^q)^{1/4}} + (T_p^{q+1})^{7/8} \le 2(T_p^{q+1})^{7/8}. \quad (5.6)$$

Now recalling that $k(q) \le \frac{q}{2^p}$, we have that

$$\sum_{p\ge 0} \sum_{q\ge p2^{p+5}} 2p_1(p,q) + p_2(p,q) + p_6(p,q) + k(q)(2p_1(p,q) + 2p_3(p,q)) \\ + (p_4(p,q) + k(q)p_5(p,q))(1 + T_p^{q+1} - T_p^q) < \infty.$$

As a result, the Borel-Cantelli lemma implies that on an event $\mathcal{E}$ of probability one, there exists $\hat{T}_1$ such that for any $p \ge 0$, $q \ge p2^{p+5}$ Eq (5.1), (5.2), (5.4) and (5.6) are satisfied, and (5.3) is satisfied for $q \ge l, p2^{p+5}$, and Eq (5.5) is satisfied for $T_p^q \le T < T_p^{q+1}$.

We are now ready to prove the universal consistency of the learning rule. First, we pose $\epsilon_p = 2\frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{2^{p/4}}$ and aim to show that the average error made by the learning rule on $\mathcal{T}_p$ is $\mathcal{O}(\epsilon_p)$ uniformly over time. Note in particular that $\sum_{p\ge 0} \epsilon_p < \infty$. For any $T \ge 1$, we define $\mathcal{R}_p(T) = \sum_{t\le T, t\in\mathcal{T}_p} r_t$ the reward obtained by the learning rule, and $\bar{R}_p^*(T) = \sum_{t\le T, t\in\mathcal{T}_p} \bar{r}(\pi^*(X_t), X_t)$ the reward obtained by the optimal policy. To do so, we first start by analyzing the regret on the first period $[1, 2^{32p})$ where there is no exploration and the learning rule uses EXP3.IX learners on each new instance. For $T < 2^{32p}$ let $\mathcal{X}_p(T) := \{X_t, t \in \mathcal{T}_p, t \le T\}$. Note that $|\mathcal{X}_p(T)| \le \frac{T}{4^p}$ by definition of $\mathcal{T}_p$. For $x \in \mathcal{X}_p(T)$, let $N_p(T;x) = |\{t \le T, t \in \mathcal{T}_p, X_t = x\}| \le 2^{2p+2}$ and $\bar{R}_p^0(T;x) := \mathbb{E}[\sum_{t\le T, t\in\mathcal{T}_p, X_t = x} \tilde{r}_t \mid \mathbb{X}]$ where $\tilde{r}_t$ is the reward obtained if we used strategy 0. Now by Theorem 5.6, for every $x \in \mathcal{X}_p(T)$, with probability at least $1 - e^{-p^2 T^{1/2^7}}$, we have

$$\sum_{t\le T, t\in\mathcal{T}_p(T), X_t = x} r_t(\pi^*(x), x) - \mathcal{R}_p(T) \le cpT^{1/2^7} \sqrt{|\mathcal{A}| \ln |\mathcal{A}| N_p(T;x)}$$

As a result, with probability at least $1 - Te^{-p^2 T^{1/2^7}} := 1 - p_7(p,T)$,

$$\begin{aligned}\sum_{t\le T, t\in\mathcal{T}_p} r_t(\pi^*(x),x) - r_t &\le \frac{2^p}{4^p}T + \sum_{x\in\mathcal{X}_p(T), N_p(T;x)\ge 2^p}\ \sum_{t\le T, t\in\mathcal{T}_p, X_t = x} r_t(\pi^*(x),x) - r_t \\ &\le \frac{T}{2^p} + cp\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1/2^7}\sum_{x\in\mathcal{X}_p(T), N_p(T;x)\ge 2^p}\sqrt{N_p(T;x)} \\ &\le \frac{T}{2^p} + cp\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1/2^7}\frac{T}{2^{p/2}} \le \frac{T}{2^p} + \frac{c}{2}\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T,\end{aligned}$$

where in the last inequality, we used $2^{2p} \le T < 2^{32p}$, thus $2^{p/2} \ge T^{1/64}$. Then, by Hoeffding's inequality, we have with probability $1 - e^{-2p^2\sqrt{T}} := 1 - p_8(p,T)$,

$$\sum_{t\le T, t\in\mathcal{T}_p} r_t(\pi^*(x),x) \ge \bar{R}_p^*(T) - \frac{\log_2 T}{2}T^{3/4}.$$

Finally, with probability at least $1 - p_7(p,T) - p_8(p,T)$, we obtain

$$\mathcal{R}_p(T) \ge \bar{R}_p^*(T) - \frac{1+c}{2}\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - \frac{T}{2^p}. \tag{5.7}$$

Noting that $\sum_{p\ge 0}\sum_{T\ge 1} p_7(p,T) + p_8(p,T) < \infty$, the Borel-Cantelli lemma implies that on an event $\mathcal{F}$ of probability one, there exists $\hat{T}_2$ such that for all $T \ge \hat{T}_2$, and $p \ge 0$ such that $T < 2^{32p}$, Eq (5.7) holds. We will now suppose that the event $\mathcal{E}\cap\mathcal{F}$ of probability one is met. Next we consider the case of $T \ge 2^{32p}$, and let $q_0 \ge p2^{p+5}$ such that $T_p^{q_0} \le T < T_p^{q_0+1}$. Let

$$\mathcal{S}_p^0 := \left\{ p2^{p+5} \le q < q_0 : \hat{R}_p^0(q) - \eta_p(T_p^{q+1} - T_p^q) \ge \max_{1\le l\le k(q)} \hat{R}_p^k(q)\right\},$$

the set of phases where the learning rule estimated that strategy 0 performed better than strategy 1. Next, let $\mathcal{P}_p^i = \{p2^{p+5} \le q < q_0 : \mathcal{P}_p(q) = i\}$ the set of phases where the learning rule performed strategy i for $i \in \{0,1\}$. An important observation is that for two phases $q_1 < q_2 \in \mathcal{S}_p^0 \cap \mathcal{P}_p^1$, if strategy 1 should not have been performed, then $q_2 > q_1 + p2^p$. In particular, we have $T_p^{q_1} \le 2^{-p}T_p^{q_2}$, hence $T_p^{q_1+1} - T_p^{q_1} \le 2^{-p}(T_p^{q_2+1} - T_p^{q_2})$. This allows to dissipate errors made during phases where the algorithm performs strategy 1 by mistake. Precisely, using a descending induction we obtain

$$\sum_{q\in\mathcal{S}_p^0\cap\mathcal{P}_p^1} T_p^{q+1} - T_p^q \le \frac{T_p^{q_0} - T_p^{q_0-1}}{1-2^{-p}} \le 2\cdot 2^{-p}T_p^{q_0} \le 2^{-p+1}T \le 2\epsilon_p T.$$

On all other phases $\mathcal{P}_p^0 \cup (\mathcal{P}_p^1 \setminus \mathcal{S}_p^0)$, the performance of the learning rule is close to having

performed strategy 0 on all phases. Indeed, using Eq (5.4) we obtain

$$\begin{aligned}\sum_{q\in(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\tilde{R}_p^1(q) &\geq \sum_{q\in(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\max_{l=1,\dots,k(q)}\bar{R}_p^l(q) - (T_p^{q+1})^{3/4} - c(T_p^{q+1})^{3/4}\ln T_p^{q+1} - A_p(q)\\ &\geq \sum_{q\in(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\max_{l=1,\dots,k(q)}\hat{R}_p^l(q) - \sum_{q<q_0}\left(4(T_p^{q+1})^{7/8} + c(T_p^{q+1})^{3/4}\ln T_p^{q+1}\right)\\ &\geq \sum_{q\in(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\hat{R}_p^0(q) - \eta_p T_p^{q_0} - 4(4 + c\ln T_p^{q_0})T^{15/16}\\ &\geq \sum_{q\in(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\bar{R}_p^*(q) - \eta_p T - 3\epsilon_p T - 4(5 + c\ln T)T^{15/16}.\end{aligned}$$

In the second inequality, we used Eq (5.3) and in the third inequality, we used the definition of $\mathcal{S}_p^0$ and the identities $\sum_{q\leq q_0}(T_p^q)^{7/8} \leq (T_p^{q_0})^{7/8}\frac{2^p}{1-2^{-7/8}} \leq 2^{p+2}(T_p^{q_0})^{7/8} \leq 4T^{15/16}$. In the last inequality, we used Eq (5.1). Next, using Eq (5.2) we have directly

$$\sum_{q\in\mathcal{P}_p^0}\tilde{R}_p^0(q) \geq \sum_{q\in\mathcal{P}_p^0}\bar{R}_p^*(q) - 3\epsilon_p T - 3\cdot 4T^{15/16}.$$

Combining the two above inequalities and observing that $\eta_p = 5\epsilon_p$ gives

$$\begin{aligned}\sum_{2^{32p}\leq t<T_p^{q_0},t\in\mathcal{T}_p} r_t &\geq \sum_{q\in\mathcal{P}_p^0}\tilde{R}_p^0(q) + \sum_{q\in\mathcal{P}_1\setminus\mathcal{S}^0}\tilde{R}_p^1(q)\\ &\geq \sum_{q\in\mathcal{P}_p^0\cup(\mathcal{P}_p^1\setminus\mathcal{S}_p^0)}\bar{R}_p^*(q) - 11\epsilon_p T - (32 + 4c\ln T)T^{15/16}\\ &\geq \sum_{p2^{p+5}\leq q<q_0}\bar{R}_p^*(q) - \sum_{q\in\mathcal{S}_p^0\cap\mathcal{P}_p^1}(T_p^{q+1} - T_p^q) - 11\epsilon_p T - (32 + 4c\ln T)T^{15/16}\\ &\geq \sum_{p2^{p+5}\leq q<q_0}\bar{R}_p^*(q) - 13\epsilon_p T - (32 + 4c\ln T)T^{15/16}.\end{aligned}$$

Now recalling the former estimate of $\mathcal{R}_p(T)$ for $T < 2^{32p}$, we obtain

$$\begin{aligned}\mathcal{R}_p(T) &\geq \mathcal{R}_p(2^{32p} - 1) + \sum_{2^{32p}\leq t<T_p^{q_0},t\in\mathcal{T}_p} r_t\\ &\geq \bar{R}_p^*(T) - 2\frac{T}{2^p} - \frac{1+c}{2}\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - 13\epsilon_p T - (32 + 4c\ln T)T^{15/16}\\ &\geq \bar{R}_p^*(T) - \frac{1+c}{2}\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - (32 + 4c\ln T)T^{15/16} - 15\epsilon_p T\end{aligned}$$

where the term $\frac{T}{2^p}$ comes from the fact that $T - (T_p^{q_0} - 1) \leq T_p^{q_0+1} - T_p^{q_0} \leq \frac{T}{2^p}$. Now note that if $t \in \mathcal{T}_p$, there were at least 4^p duplicates, hence $t \geq 4^p$. As a result, we can always suppose without loss of generality that $T \geq 4^p$. Combining with the case $T < 2^{32p}$, we obtain that for all $T \geq \max(\hat{T}_1, \hat{T}_2)$, $p \geq 0$ with $t \geq 4^p$,

$$\mathcal{R}_p(T) \geq \bar{R}_p^*(T) - (33 + 5c)\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - 15\epsilon_p T. \tag{5.8}$$

This ends the proof that on times $\mathcal{T}_p$, the learning rule has an average error at most $\mathcal{O}(\epsilon_p)$ on the event $\mathcal{E} \cap \mathcal{F}$. Because $\sum_{p\geq 0} \epsilon_p < \infty$, we can afford to converge on each set $\mathcal{T}_p$ to the optimal policy independently. Fix $0 < \epsilon \leq 1, \delta > 0$ and let p_0 such that $\sum_{p\geq p_0} \epsilon_p < \frac{\epsilon}{15}$. Because $\mathbb{X} \in \text{SMV}$, by Proposition 5.1, $\mathbb{X}^{\leq 4^{p_0}} \in \text{CS}$. As a result, because the sequence of policies $(\pi^l)_l$ is dense under CS processes, there exists $l_0 \geq 1$ such that

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)]\right] \leq \frac{\epsilon\delta}{2^{2p_0+2}p_0}.$$

Then, by the dominated convergence theorem, there exists T_0 such that

$$\mathbb{E}\left[\sup_{T\geq T_0} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)]\right] \leq \frac{\epsilon\delta}{2^{2p_0+1}p_0}.$$

Thus, on an event $\mathcal{B}_\delta$ of probability at least $1-\delta$, by the Markov inequality, for all $T \geq T_0$,

$$\sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)] \leq \frac{\epsilon}{2^{2p_0+1}p_0} T.$$

In particular, the above equation holds if we replace $\mathcal{T}^{\leq 4^{p_0}}$ by $\mathcal{T}_p$ for any $p < p_0$. Now suppose that the event $\mathcal{E} \cap \mathcal{F} \cap \mathcal{B}_\delta$ of probability at least $1-\delta$ is met. For any $p < p_0$ and $q \geq p2^{p+5}$ such that $T_p^q \geq \hat{T} := \max(\hat{T}_1, \hat{T}_2, 2^{l_0}, 2^{32p_0})$, because $T_p^q \geq 2^{l_0}$, we have

$$\begin{aligned}
\max_{1\leq l\leq k(q)} \hat{R}_p^k(q) \geq \hat{R}_p^{l_0}(q) &\geq \bar{R}_p^{l_0}(q) - (T_p^{q+1})^{7/8} \\
&\geq \bar{R}_p^*(q) - (T_p^{q+1})^{7/8} - \sum_{t\in\mathcal{T}_p(q)} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)] \\
&\geq \hat{R}_p^0(q) - 2(T_p^{q+1})^{7/8} - 3\epsilon_p(T_p^{q+1} - T_p^q) - 2^{-2p-1}T_p^{q+1} \\
&\geq \hat{R}_p^0(q) - 2(T_p^{q+1})^{7/8} - 4\epsilon_p(T_p^{q+1} - T_p^q).
\end{aligned}$$

where in the second inequality we used Eq (5.3) and in the fourth we used Eq (5.1). In the last inequality, we used $2^{-p-1}T_p^{q+1} \leq T_p^{q+1} - T_p^q$. Now let T_1 such that $2T^{7/8} < \frac{\epsilon_p}{2^{p+1}}T$ for any $T \geq T_1$. Then, for any $p < p_0$ and $q \geq p2^{p+5}$ such that $T_p^q \geq \tilde{T} := \max(\hat{T}, T_1)$, we have

$$\max_{1\leq l\leq k(q)} \hat{R}_p^k(q) > \hat{R}_p^0(q) - 5\epsilon_p(T_p^{q+1} - T_p^q),$$

which implies $\mathcal{P}_p(q+1) = 1$ since $\eta_p = 5\epsilon_p$ if $\mathcal{P}_p(q+1)$ was not already defined. In other terms, starting from time $2^{p_0}\tilde{T}$, the learning rule always chooses strategy 1 for categories $p < p_0$. We now bound the error of the learning rule on $\mathcal{T}_p$ for $p < p_0$. Let $\tilde{q}$ such that

$T_p^{\tilde{q}-1} \leq 2^{p_0}\tilde{T} < T_p^{\hat{q}}$. For any $T \geq 2^{p_0}\tilde{T}$ and $q(T)$ such that $T_p^{q(T)} \leq T < T_p^{q(T)+1}$, we can write

$$\begin{aligned}
\mathcal{R}_p(T) - \bar{R}_p^*(T) &\geq \sum_{\tilde{q}<q<q(T)} (\tilde{R}_p^1(q) - \bar{R}_p^*(q)) + \tilde{R}_p^1(q(T), T) - \sum_{t\in\mathcal{T}_p(q), t\leq T} \bar{r}(\pi^*(X_t), X_t) \\
&\qquad - 2^{p_0}\tilde{T} - \sum_{q<q(T)} A_p(q) \\
&\geq \sum_{\tilde{q}<q<q(T)} (R_p^{l_0}(q) - \bar{R}_p^*(q)) - \sum_{t\in\mathcal{T}_p(q), t\leq T} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)] - 2^{p_0}\tilde{T} \\
&\qquad - \sum_{q\leq q(T)} (2A_p(q) + (T_p^{q+1})^{3/4} + c(T_p^{q+1})^{3/4} \ln T_p^{q+1}) \\
&\geq - \sum_{t\leq T, t\in\mathcal{T}_p} \mathbb{1}[\pi^*(X_t) \neq \pi^{l_0}(X_t)] - 2^{p_0}\tilde{T} - 4(3+c)(T_p^{q(T)+1})^{15/16} \ln T_p^{q(T)+1} \\
&\geq -2^{p_0}\tilde{T} - 16(3+c)T^{15/16} \ln T - \frac{\epsilon}{2^{p_0}p_0} T.
\end{aligned}$$

where in the second inequality we applied Eq (5.4) and Eq (5.5), and in the third inequality, we used $\sum_{q\leq q(T)} (T_p^{q+1} - T_p^q)^{3/4} \leq 4(T_p^{q(T)+1})^{7/8}$ proved earlier. As a result, we can write

$$\sum_{p<p_0} \bar{R}_p^*(T) - \mathcal{R}_p(T) \leq p_0 2^{p_0}\tilde{T} + 16p_0(3+c)T^{15/16} \ln T + \epsilon T.$$

Now because the events $\mathcal{E}, \mathcal{F}$ are met, using Eq (5.8), we also have for $T \geq \tilde{T}$

$$\begin{aligned}
\sum_{p\geq p_0} \bar{R}_p^*(T) - \mathcal{R}_p(T) &= \sum_{p_0\leq p<\log_4 T} \bar{R}_p^*(T) - \mathcal{R}_p(T) \\
&\leq (17+3c)\sqrt{|\mathcal{A}| \ln |\mathcal{A}|} T^{1-1/2^7} (\log_2 T)^2 + 15 \sum_{p\geq p_0} \epsilon_p \cdot T \\
&\leq (17+3c)\sqrt{|\mathcal{A}| \ln |\mathcal{A}|} T^{1-1/2^7} (\log_2 T)^2 + \epsilon T
\end{aligned}$$

Summing the two above inequalities gives

$$\begin{aligned}
&\sum_{t=1}^T \bar{r}(\pi^*(X_t), X_t) - r_t \\
&\qquad \leq p_0 2^{p_0}\tilde{T} + 16p_0(3+c)T^{15/16} \ln T + (17+3c)\sqrt{|\mathcal{A}| \ln |\mathcal{A}|} T^{1-1/2^7} (\log_2 T)^2 + 2\epsilon T.
\end{aligned}$$

As a result, on the event $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} \cap \mathcal{B}_\delta$ of probability at least $1-\delta$, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \bar{r}(\pi^*(X_t), X_t) - r_t \leq 2\epsilon.$$

Because this holds for any $\delta > 0$ and $0 < \epsilon < 1$, this shows that almost surely, we have $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \bar{r}(\pi^*(X_t), X_t) - r_t \leq 0$. We denote by $\mathcal{D}$ this event. We now formally

show that the learning rule is universally consistent. Let $\pi : \mathcal{X} \to \mathcal{A}$ be a measurable function. First, by the Hoeffding inequality, we have for $T \geq 1$,

$$\mathbb{P}\left[\left|\sum_{t=1}^{T} r_t(\pi(X_t), X_t) - \bar{r}_t(\pi(X_t), X_t)\right| \leq T^{3/4}\right] 1 - e^{-2\sqrt{T}}.$$

As a result, the Borel-Cantelli lemma implies that on an event $\mathcal{H}$ of probability one, there exists $\hat{T}_4$ such that for all $T \geq \hat{T}_4$, $|\sum_{t=1}^{T} r_t(\pi(X_t), X_t) - \bar{r}_t(\pi(X_t), X_t)| \leq T^{3/4}$. Then, on $\mathcal{D} \cap \mathcal{H}$ of probability one, for any $T \geq \hat{T}_4$ we have

$$\sum_{t=1}^{T} r(\pi(X_t), X_t) - r_t \leq \sum_{t=1}^{T} \bar{r}(\pi(X_t), X_t) - r_t + T^{3/4} \leq \sum_{t=1}^{T} \bar{r}(\pi^*(X_t), X_t) - r_t + T^{3/4}.$$

Thus, $\limsup_{T\to\infty} \sum_{t=1}^{T} \bar{r}(\pi(X_t), X_t) - r_t \leq 0$. This ends the proof that the learning rule is universally consistent under any SMV process. Now recall that SMV is a necessary condition for universal learning by Theorem 5.7. Hence, the set of learnable processes is exactly SOCB = SMV and the learning rule is optimistically universal. ■

5.5 Countably Infinite Action Spaces

We next turn to the case where the action space is infinite $|\mathcal{A}| = \infty$ but countable. The goal of this section is to show that the set of processes admitting universal learning now becomes CS. This contrasts with the full-feedback setting where in Chapter 4, we showed that universal learning is optimistically achievable under SMV processes when a property F-TiME on the value space $(\mathcal{Y}, \ell)$ is satisfied. Intuitively, this asks that mean-estimation is possible in finite time for any prescribed error tolerance. Of interest to the discussion of this section for a countable number of actions, we previously showed that countably-infinite classification $(\mathcal{Y}, \ell) = (\mathbb{N}, \ell_{01})$ satisfies the F-TiME property and, their learning rule is universally consistent under SMV processes even under noisy and adversarial responses.

For countable action sets, there is a simple optimistically universal learning rule defined as follows. From [Han21a, Lemma 24], because $\mathcal{A}$ is countable, the $0-1$ loss on $\mathcal{A}$ is a separable metric, thus, there exists a dense countable set $\Pi = (\pi^l)_{l\geq 1}$ of measurable policies as described in Lemma 5.2. For every $\mathbb{X} \in$ CS and every measurable $\pi^\star : \mathcal{X} \to \mathcal{A}$, we have

$$\mathbb{E}\left[\inf_{\pi\in\Pi} \hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^\star(x)\})\right] \leq \inf_{\pi\in\Pi} \mathbb{E}\left[\hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^\star(x)\})\right] = 0,$$

which implies in particular that almost surely, $\inf_{\pi\in\Pi} \hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^\star(x)\}) = 0$. For any $\mathbb{X}$, we consider the countable set of experts $\{E_1, E_2, \ldots\}$ such that $E_{i,t} = \pi_i(X_t)$. Our learning rule then applies EXPINF from Corollary 5.1 with this family of experts.

Theorem 5.9. *Let $\mathcal{X}$ be a separable Borel-metrizable space and $\mathcal{A}$ a countable infinite action set. Then, there is an optimistically universal learning rule and the set of learnable processes is SOCB = CS.*

Proof We start by showing that the learning rule defined above is universally consistent on any $\mathbb{X} \in$ CS process. This proof is essentially identical to that of [Han22, Theorem 1]. Indeed, denoting by $\hat{a}_t$ the action selected by the learning rule at time t, Corollary 5.1 implies that on an event $\mathcal{E}$ of probability one, for any $\pi \in \Pi$, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - r_t(\hat{a}_t) \le 0.$$

Now fix a measurable policy $\pi^\star : \mathcal{X} \to \mathcal{A}$. For any $\pi \in \Pi$, because the rewards lie in $[0,1]$, on $\mathcal{E}$,

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \\
&\le \hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^*(x)\}) + \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - r_t(\hat{a}_t) \\
&\le \hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^*(x)\}).
\end{aligned}$$

Also, by construction of the countable set Π, on an event $\mathcal{F}$ of probability one, we have $\inf_{\pi\in\Pi} \hat{\mu}_{\mathbb{X}}(\{x : \pi(x) \neq \pi^\star(x)\}) = 0$. Thus, on $\mathcal{E} \cap \mathcal{F}$, the above inequality shows that $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \le 0$. Hence, the learning rule is universally consistent under CS processes with adversarial responses.

Next, we show that the condition $\mathbb{X} \in$ CS is necessary for the existence of a universally consistent learning rule, even for function learning. Let $\mathbb{X}$ be any process with $\mathbb{X} \notin$ CS. By Lemma 5.1, there exists a sequence $\{B_i\}_{i=1}^{\infty}$ of disjoint measurable subsets of $\mathcal{X}$ with $\bigcup_{i\in\mathbb{N}} B_i = \mathcal{X}$, and a sequence $\{N_i\}_{i=1}^{\infty}$ in $\mathbb{N}$ such that, on a $\sigma(\mathbb{X})$-measurable event $\mathcal{E}_0$ of probability strictly greater than zero,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] > 0,$$

where i_t is the unique $i \in \mathbb{N}$ with $X_t \in B_i$.

Next, we define the function $f^\star$. Enumerate $\mathcal{A} = \{a_1, a_2, \ldots\}$, and for each $i \in \mathbb{N}$, let $A_i = \{a_1, \ldots, a_{2N_i}\}$. For each $i \in \mathbb{N}$, let $a_i^\star$ be an element of A_i. Denote by $\bar{a} = \{a_i^\star\}_{i\in\mathbb{N}}$. Then for each $i \in \mathbb{N}$ and each $x \in B_i$, define $f^\star_{\bar{a}}(x,a) = \mathbb{1}[a = a_i^\star]$. Also define $\mathbf{a}_i^\star$ as Uniform(A_i) (independent over i and all independent of $\mathbb{X}$ and the randomness of the learning rule), and $\bar{\mathbf{a}} = \{\mathbf{a}_i^\star\}_{i\in\mathbb{N}}$. Then for any learning rule $\hat{f}_t$, denoting by $\hat{a}_t$ its actions when $f^\star = f^\star_{\bar{\mathbf{a}}}$ is as constructed above, we have

$$\begin{aligned}
\sup_{\bar{a}} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\left(\sup_{a\in\mathcal{A}} r_t(a) - r_t(\hat{a}_t)\right)\middle|\bar{\mathbf{a}} = \bar{a}\right] &= \sup_{\bar{a}} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\hat{a}_t \neq \mathbf{a}_{i_t}]\middle|\bar{\mathbf{a}} = \bar{a}\right] \\
&\ge \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\hat{a}_t \neq \mathbf{a}_{i_t}]\right] \\
&\ge \mathbb{E}\left[\mathbb{1}_{\mathcal{E}_0} \cdot \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}]\, \mathbb{1}[\hat{a}_t \neq \mathbf{a}_{i_t}]\right].
\end{aligned}$$

By the law of total expectation, this last expression above equals

$$\mathbb{E}\left[\mathbb{1}_{\mathcal{E}_0} \cdot \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}]\, \mathbb{1}[\hat{a}_t \neq \mathbf{a}_{i_t}] \middle| \mathbb{X}, \hat{f}_\cdot\right]\right],$$

where conditioning on $\hat{f}_\cdot$ indicates we condition on the independent randomness of the learning rule. Since the average is bounded for any fixed T, Fatou's lemma, together with the fact that $\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}]$ is $\sigma(\mathbb{X})$-measurable, imply the expression above is at least

$$\mathbb{E}\left[\mathbb{1}_{\mathcal{E}_0} \cdot \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}]\, \mathbb{P}\left(\hat{a}_t \neq \mathbf{a}_{i_t} \middle| \mathbb{X}, \hat{f}_\cdot\right)\right]. \tag{5.9}$$

Let $\hat{N}_t = |\mathbb{X}_{\leq t} \cap B_{i_t}|$ and $\hat{A}_t = \{\hat{a}_{t'} : t' \leq t, i_{t'} = i_t\}$. Note that, conditioned on $\hat{f}_\cdot$ and $\mathbb{X}$, the probability that $\mathbf{a}_{i_t} \in \hat{A}_t$ is at most $\hat{N}_t \frac{1}{|A_{i_t}|} = \frac{\hat{N}_t}{2N_{i_t}}$. In particular, if $\hat{N}_t \leq N_{i_t}$, the conditional probability (given $\hat{f}_\cdot$ and $\mathbb{X}$) that $\hat{a}_t \neq \mathbf{a}_{i_t}$ is at least $1 - \frac{\hat{N}_t}{2N_{i_t}} \geq \frac{1}{2}$. Thus, (5.9) is no smaller than

$$\mathbb{E}\left[\mathbb{1}_{\mathcal{E}_0} \cdot \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] \cdot \frac{1}{2}\right]. \tag{5.10}$$

By definition of the event $\mathcal{E}_0$, there is a nonzero probability that

$$\mathbb{1}_{\mathcal{E}_0} \cdot \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] > 0,$$

and since the quantity on the left-hand side is non-negative, this further implies the expectation in (5.10) is also strictly greater than zero.

Altogether, this implies there exists a (non-random) choice of $\bar{a}$ such that, choosing $f^\star = f^\star_{\bar{a}}$, the actions $\hat{a}_t$ made by the learning rule $\hat{f}_t$ satisfy

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} r_t(a) - r_t(\hat{a})\right)\right] > 0,$$

and since the quantity in the expectation is non-negative, this further implies that for this choice of $f^\star$, with non-zero probability,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} r_t(a) - r_t(\hat{a})\right) > 0.$$

Thus, $\hat{f}_t$ is not universally consistent for function learning. Since this holds for any choice of learning rule $\hat{f}_\cdot$, this completes the proof. ■

5.6 Uncountable Action Spaces

We next consider the case of uncountable action spaces. In this section, we assume that $\mathcal{A}$ is an uncountable separable Borel metrizable space. In this case, we will show that universal consistency is impossible even in the simplest setting where rewards are deterministic, i.e., $r_t(a) = f^*(X_t, a)$ for some unknown measurable function $f^* : \mathcal{X} \times \mathcal{A} \to [0, 1]$. The argument is based on a dichotomy depending on whether there exists a non-atomic probability measure μ on $\mathcal{A}$, i.e., such that for all $a \in \mathcal{A}$, we have $\mu(\{x\}) = 0$. If this is not the case, we will need the following simple result which states that any stochastic process $\mathbb{X}$ takes values in a countable set $Supp(\mathbb{X})$ almost surely.

Lemma 5.3. *Let $\mathcal{X}$ be a metrizable separable Borel space such that there does not exist a non-atomic probability measure on $\mathcal{X}$. Then, for any random variable X on $\mathcal{X}$ there exists a countable set $Supp(X) \subset \mathcal{X}$ such that almost surely, $X \in Supp(X)$. Similarly, for any stochastic process $\mathbb{X}$ on $\mathcal{X}$ there exists a countable set $Supp(\mathbb{X}) \subset \mathcal{X}$ such that almost surely $\forall t \geq 1, X_t \in Supp(\mathbb{X})$.*

Proof Fix $\mathcal{X}$ such a space and let X be a random variable on $\mathcal{X}$. Let $Supp(X) = \{x \in \mathcal{X} : \mathbb{P}[X = x] > 0\}$. Suppose by contradiction that $\mathbb{P}[X \notin Supp(X)] > 0$ and denote $\mathcal{E}$ the corresponding event. Because $\mathbb{P}[\mathcal{E}] > 0$ we can consider a random variable $Y \sim X|\mathcal{E}$. For instance take $(X_i)_{i\geq 1}$ an i.i.d. process following the distribution of X, fix $x_0 \in \mathcal{X}$ a fixed arbitrary instance, and pose

$$Y = \begin{cases} X_{\hat{k}} & \text{if } \{i \geq 1 : X_i \notin Supp(X)\} \neq \emptyset, \quad \hat{k} = \min\{i \geq 1 : X_i \notin Supp(X)\}, \\ x_0 & \text{otherwise.} \end{cases}$$

Because the first time k such that $X_k \notin Supp(X)$ is a geometric variable $\mathcal{G}(1 - \mathbb{P}[\mathcal{E}])$, the event $\mathcal{F} = \{\exists i \geq 1 : X_i \notin Supp(X)\}$ has probability one. We now show that Y is non-atomic. First, observe that $Y \notin Supp(X)$. Then, if $x \in \mathcal{X} \notin Supp(X)$, we have

$$\mathbb{P}[Y = x] = \mathbb{P}[\{Y = x\} \cap \mathcal{F}] = \mathbb{P}[\{X_{\hat{k}} = x\} \cap \mathcal{F}] \leq \mathbb{P}\left[\bigcup_{i\geq 1}\{X_i = x\}\right] \leq \sum_{i\geq 1} \mathbb{P}[X_i = x] = 0.$$

where in the first equality we used the fact that $\mathbb{P}[\mathcal{F}^c] = 0$. Therefore Y is non-atomic which contradicts the hypothesis on $\mathcal{X}$. As a result, almost surely $X \in Supp(X)$. It now suffices to check that $Supp(X)$ is countable, which is guaranteed by the identity $1 = \mathbb{P}[X \in Supp(X)] = \sum_{x\in Supp(X)} \mathbb{P}[X = x]$, since each term of the sum is positive. This ends the proof of the first claim.

Now let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$ and define $Supp(\mathbb{X}) = \bigcup_{t\geq 1} Supp(X_t)$. Then $Supp(\mathbb{X})$ is countable as a countable union of countable sets and

$$\mathbb{P}[\exists t \geq 1 : X_t \notin Supp(\mathbb{X})] \leq \sum_{t\geq 1} \mathbb{P}[X_t \notin Supp(\mathbb{X})] \leq \sum_{t\geq 1} \mathbb{P}[X_t \notin Supp(X_t)] = 0.$$

This ends the proof of the lemma. ■

We are now ready to show that no process admits universal learning for uncountable action sets.

Theorem 5.10. *If $\mathcal{A}$ is an uncountable separable Borel metrizable space, then there does not exist any $\mathbb{X}$ admitting universal consistency for measurable function learning.*

Proof Fix a learning rule $f.$ and for any $a^* \in \mathcal{A}$, we define the reward function $f^*_{a^*}(x, a) = \mathbb{1}[a = a^*]$ for $x \in \mathcal{X}, a \in \mathcal{A}$. We also define the policy $\pi_{a^*} : x \in \mathcal{X} \mapsto a^* \in \mathcal{A}$. We first consider the case where there exists a non-atomic probability measure μ on $\mathcal{A}$. Then, for any $t \geq 1$, and consider the case where a^* is sampled from the distribution μ independently from the process $\mathbb{X}$ and the randomness of the learning rule. Then we have

$$\mathbb{P}_{a^* \sim \mu}[f_t(\mathbb{X}_{<t}, (0)_{<t}, X_t) = a^*] = \mathbb{E}_{\mathbb{X}, f_t}[\mathbb{P}_{a^* \sim \mu}(f_t(\mathbb{X}_{<t}, (0)_{<t}, X_t) = a^*)] = 0.$$

Denote by $\mathcal{E}_t$ this event. Then, by the union bound, $\mathbb{P}[\bigcap_{t \geq 1} \mathcal{E}_t] = 1$. The law of total probability implies that there exists a deterministic choice of a^* such that

$$\mathbb{P}[\forall t \geq 1, f_t(\mathbb{X}_{<t}, (0)_{<t}, X_t) \neq a^*] = 1,$$

where the probability is taken over $\mathbb{X}$ and the randomness of the learning rule.

Now suppose that there does not exist non-atomic probability measures on $\mathcal{A}$. From Lemma 5.3, for any probability measure μ on $\mathcal{A}$, we can construct a countable set $Supp(\mu) \subset \mathcal{X}$ such that $\mu(Supp(\mu)) = 1$. Now consider the set

$$S = \bigcup_{t \geq 1} Supp(f_t(\mathbb{X}_{\leq t-1}, (0)_{\leq t-1}, X_t)).$$

Then, S is countable as the union of countable sets. Since $\mathcal{A}$ is uncountable, let $a^* \in \mathcal{A} \backslash S$. By construction, on an event of probability one, for all $t \geq 1$, we have $f_t(\mathbb{X}_{\leq t-1}, (0)_{\leq t-1}, X_t) \neq a^*$.

In both cases, we found an action $a^* \in \mathcal{A}$ such that on an event $\mathcal{E}$ of probability one over $\mathbb{X}$ and the randomness of the learning rule, having received 0 reward in the past history at time step t, the learning rule does not select a^*, hence receives reward 0 at time t as well. Thus, by induction, denoting by $\hat{a}_t$ the action selected by the learning rule at time t for reward $f^*_{a^*}$, we have $\mathcal{E} \subset \{\forall t \geq 1, \hat{a}_t \neq a^*\}$. Thus, on $\mathcal{E}$,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) = 1.$$

Because $\mathcal{E}$ has probability one, this shows that $f.$ is not universally consistent. ■

5.7 Universal Learning under Continuity Assumptions

In Section 5.6 we showed that for general uncountable separable metric action spaces, without further assumptions on the rewards, one cannot achieve universal consistency. The goal of this section is to show that adding mild continuity assumptions on the rewards enables universal consistency under a large family of processes.

5.7.1 Continuous rewards

In this section, we suppose that the rewards are continuous as defined in Definition 5.2, and show that universal consistency on CS processes is still achievable. For bounded separable metric action spaces $(\tilde{\mathcal{A}}, \tilde{d})$, [Han21a] showed that there is countable set of measurable policies Π such that for any measurable $\pi^* : \mathcal{X} \to \tilde{\mathcal{A}}$ and $\mathbb{X} \in \text{CS}$,

$$\inf_{\pi \in \Pi} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \tilde{d}(\pi^*(X_t), \pi(X_t))\right] = 0.$$

In general, the action space $(\mathcal{A}, d)$ is unbounded, however, $(\mathcal{A}, d \wedge 1)$ is a separable bounded metric space on which we can apply the above result. This provides a countable set of measurable policies Π such that for any measurable $\pi^* : \mathcal{X} \to \mathcal{A}$ and $\mathbb{X} \in \text{CS}$,

$$\inf_{\pi \in \Pi} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \tilde{d}(\pi^*(X_t), \pi(X_t)) \wedge 1\right] = 0.$$

From this observation, we can get the following lemma.

Lemma 5.4. *Let $\mathcal{X}$ be a separable metrizable Borel space and $(\mathcal{A}, d)$ be a separable metric space. For any measurable function $\pi^* : \mathcal{X} \to \mathcal{A}$, on an event of probability one, for all $i \geq 1$, there exists $\pi^i \in \Pi$ such that*

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[d(\pi^*(X_t), \pi^i(X_t)) \geq 2^{-i}] \leq 2^{-i},$$

and for all $\pi \in \{\pi^i : i \geq 1\} \cup \{\pi^*\}$, $\frac{1}{T}\sum_{t\leq T} r_t(\pi(X_t)) - \bar{r}_t(\pi(X_t)) \to 0$.

Proof By construction of the countable set of policies Π, for any $i \geq 1$, there exists $\pi^i \in \Pi$ such that

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d(\pi^*(X_t), \pi^i(X_t)) \wedge 1\right] \leq 2^{-3i}.$$

Then, Markov's inequality implies that with probability at least $1 - 2^{-i}$.

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d(\pi^*(X_t), \pi^i(X_t)) \wedge 1 \leq 2^{-2i}.$$

Applying Markov's inequality a second time, we obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[d(\pi^*(X_t), \pi^i(X_t)) \geq 2^{-i}] \leq 2^i \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d(\pi^*(X_t), \pi^i(X_t)) \wedge 1 \leq 2^{-i}.$$

The Borel-Cantelli lemma implies that on an event $\mathcal{E}$ of probability one, for i sufficiently large, there exists $\pi^i \in \Pi$ with $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[d(\pi^*(X_t), \pi^i(X_t)) \geq 2^{-i}] \leq 2^{-i}$. This

implies that this is the case for all $i \geq 1$. For any $i \geq 1$, Azuma's inequality implies that with probability at least $1 - 4e^{-2i\sqrt{T}}$, we have

$$\left|\sum_{t=1}^{T} r_t(\pi^i(X_t)) - \bar{r}_t(\pi^i(X_t))\right|, \left|\sum_{t=1}^{T} r_t(\pi^*(X_t)) - \bar{r}_t(\pi^*(X_t))\right| \leq 2iT^{3/4}.$$

Because $\sum_{T\geq 1}\sum_{i\geq 1} e^{-2i\sqrt{T}} < \infty$, the Borel-Cantelli lemma implies that on an event $\mathcal{F}$ of probability one, for all $i \geq 1$, $\frac{1}{T}\sum_{t\leq T} r_t(\pi^i(X_t)) - \bar{r}_t(\pi^i(X_t)) \to 0$ and similarly for π^*. Therefore, on the event $\mathcal{E} \cap \mathcal{F}$ of probability one, all events are satisfied, which ends the proof of the lemma. ■

Using Lemma 5.4, we will show that the EXPINF algorithm over the set of policies Π is optimistically universal for continuous rewards.

Theorem 5.11. *Let $(\mathcal{A}, d)$ be an infinite separable metric space. Then,* EXPINF *is optimistically universal for continuous rewards, and the set of learnable processes* SOCB-C = CS *for continuous rewards.*

Proof We start by showing that EXPINF is universally consistent under continuous rewards under CS processes. Let $\mathbb{X} \in$ CS and continuous rewards $(r_t)_t$ and let $\pi^* : \mathcal{X} \to \mathcal{A}$ be measurable policy. We denote $\mathcal{E}$ the event on which the guarantee for EXPINF of Corollary 5.1 holds. For convenience, we also note $\hat{a}_t$ the action selected by the learning rule at time t. For any $x \in \mathcal{X}$, and $\epsilon > 0$, we define

$$\Delta_\epsilon(x) = \sup_{a\in\mathcal{A}: d(a,\pi^*(x))\leq\epsilon} |\bar{r}(a, x) - \bar{r}(\pi^*(x), x)|.$$

Next, fix $\delta > 0$, and for any $\epsilon > 0$, let $A(\epsilon, \delta) = \{x \in \mathcal{X} : \Delta_\epsilon(x) \geq \delta\}$. Note that for any $x \in \mathcal{X}$, by continuity of $\bar{r}(\cdot, x)$, for any $\delta > 0$, $\bigcap_{\epsilon>0} A(\epsilon, \delta) = \emptyset$. By Lemma 5.4, on an event $\mathcal{F}$ of probability one, for any $i \geq 1$, there exists $\pi^i \in \Pi$ such that

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[d(\pi^*(X_t), \pi^i(X_t) \geq 2^{-i}] \leq 2^{-i},$$

$\frac{1}{T}\sum_{t\leq T} r_t(\pi^i(X_t) - \bar{r}_t(\pi^i(X_t)) \to 0$ and similarly for π^*. As a result, on $\mathcal{F}$, for any $i \geq 1$,

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T}\sum_{t\leq T} r_t(\pi^*(X_t), X_t) - r_t(\pi^i(X_t), X_t) \\
&\leq \hat{\mu}_{\mathbb{X}}(A(2^{-i}, \delta)) + 2^{-i} + \limsup_{T\to\infty} \frac{1}{T} \sum_{\substack{t\leq T, \\ d(\pi^*(X_t),\pi^i(X_t))<2^{-i} \\ \Delta_{2^{-i}}(X_t)<\delta}} \bar{r}_t(\pi^*(X_t), X_t) - \bar{r}_t(\pi^i(X_t), X_t) \\
&\leq \hat{\mu}_{\mathbb{X}}(A(2^{-i}, \delta)) + 2^{-i} + \delta.
\end{aligned}$$

Because $\mathbb{X} \in$ CS and $A(2^{-i}, \delta) \downarrow \emptyset$, on an event $\mathcal{G}(\delta)$ of probability one, we have that $\hat{\mu}_{\mathbb{X}}(A(2^{-i}, \delta)) \underset{i\to\infty}{\longrightarrow} 0$. Last, let $\delta_j = 2^{-j}$ for any $j \geq 0$. On the event $\mathcal{E} \cap \mathcal{F} \cap \bigcap_{j\geq 0} \mathcal{G}(\delta_j)$ of

probability one, combining Corollary 5.1 together with the above inequality implies that for any $j \geq 0$,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T} r_t(\pi^*(X_t), X_t) - r_t(\hat{a}_t, X_t) \leq \delta_j.$$

Thus, $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T} r_t(\pi^*(X_t), X_t) - r_t(\hat{a}_t, X_t) \leq 0$ $(a.s.)$, which shows that EXPINF is universally consistent under $\mathbb{X}$. This ends the proof of the theorem.

We now show that CS is necessary for universal consistency. The proof is analogous to that of Theorem 5.9 in which we proved that for unrestricted rewards on countably infinite action sets, CS is necessary for universal learning. Suppose that $\mathbb{X} \in \text{CS}$ and let $f.$ be a learning rule. Using the same arguments, there exist a partition of $\mathcal{X}$ in measurable sets $\{B_i\}_{i\geq 1}$ and a sequence $\{N_i\}_{i\geq 1}$ of integers such that with non-zero probability,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] > 0,$$

where i_t is the index such that $X_t \in B_i$. As in the original proof, let $\{a_i, i \geq 1\}$ be a sequence of distinct actions and let $A_i = \{a_1, \ldots, a_{2N_i}\}$ for $i \geq 1$. We also define $\epsilon_i = \min_{a\neq a'\in A_i} d(a, a')$ the minimum distance within A_i actions. For any sequence $\bar{a} = \{a_i^*\}_{i\in\mathbb{N}}$ where $a_i^* \in A_i$ for $i \geq 1$, we define a deterministic reward $r_{\bar{a}}^*$ with

$$r_{\bar{a}}^*(a, x) = \max\left(1 - \frac{2d(a, a_i^*)}{\epsilon_i}, 0\right),$$

for any $x \in B_i$, which defines a proper measurable continuous reward. We also define the rewards $\tilde{r}_{\bar{a}}^*(a, x) = \mathbb{1}[a = a_i^*]$ for $a \in \mathcal{A}$ and $x \in B_i$. We now define the learning rule $\tilde{f}.$ which at each step t computes the action $\hat{a}$ chosen by the learning rule $f.$, selects the action $\tilde{a}_t := \arg\min_{a'\in A_i} d(\hat{a}, a')$ where $i \geq 1$ is the unique index with $X_t \in B_i$, receives a reward r_t, then reports the reward $\max\left(1 - \frac{2d(\hat{a},\tilde{a})}{\epsilon_i}, 0\right)$, which will be then used by $f.$ for future action selections. Note that on B_i, the rewards $r_{\bar{a}}^*$ were defined so that they are identically zero outside of the balls $B_d(a, \epsilon_i)$ for $a \in A_i$. These are disjoint, so the report of reward given by $\tilde{f}.$ to its internal run of $f.$ coincides exactly with what $f.$ would have received by selecting action $\hat{a}$ instead of $\tilde{a}$. Further, one can observe that selecting one of the nearest elements within A_i always increases the reward because the balls $B_d(a, \epsilon_i)$ for $a \in A_i$ are disjoint. Therefore, $\tilde{f}.$ always receives a higher reward than $f.$ at any step. Now observe that $\tilde{f}.$ always observes a reward in $\{0, 1\}$. Hence, for any choice of $\bar{a}$, at any step t, $\tilde{f}_t$ has the same rewards on $r_{\bar{a}}^*$ as it would have obtained on the rewards $\tilde{r}_{\bar{a}}^*$. Therefore,

$$\begin{aligned}\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} r_{\bar{a},t}^*(a) - r_{\bar{a},t}^*(\hat{a}_t)\right) &\geq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} r_{\bar{a},t}^*(a) - r_{\bar{a},t}^*(\tilde{a}_t)\right) \\ &= \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} \tilde{r}_{\bar{a},t}^*(a) - \tilde{r}_{\bar{a},t}^*(\tilde{a}_t)\right),\end{aligned}$$

where $\hat{a}_t$ (resp. $\tilde{a}_t$) denotes the action selected by $f.$ (resp. $\tilde{f}.$) at time t. However, the proof of Theorem 5.9 precisely shows that there exists a choice of $\bar{a}$ such that with non-zero probability, $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sup_{a\in\mathcal{A}} \tilde{r}_{\bar{a},t}^*(a) - \tilde{r}_{\bar{a},t}^*(\tilde{a}_t)\right) > 0$. Now observe that the

measurable function $\pi(x) = a_i^*$ where $x \in B_i$ always selects the best action. This shows that $f_\cdot$ is not consistent on rewards $r_{\tilde{a}}^*$, hence not universally consistent. Thus, $\mathbb{X} \notin$ SOCB-C which completes the proof of the theorem. ■

5.7.2 Uniformly-continuous rewards

In the last section, we showed that adding a continuity constraint on the rewards allowed us to learn CS processes even when the action space $\mathcal{A}$ is infinite. Unfortunately, this additional assumption on the rewards is not sufficient to obtain universal consistency on the more general class of processes SMV. In this section, we strengthen the assumptions on the rewards, supposing they are *uniformly-continuous* in the actions (Definition 5.2).

We start by giving necessary conditions for uniformly-continuous rewards. To do so, we will need the following simple reduction, showing that some necessary conditions provided in the unrestricted rewards case can be used in the uniformly-continuous setting as well.

Lemma 5.5. *Let $(\mathcal{A}, d)$ be a separable metric space. Let $S \subset \mathcal{A}$ with $\min_{a,a' \in S} d(a, a') > 0$. Then, SOCB-UC$(\mathcal{A}) \subset$ SOCB(S).*

Proof Intuitively, we restrict the problem on $\mathcal{A}$ to the actions S. Formally, let $\eta = \frac{1}{3}\min_{a,a' \in S} d(a, a')$ and observe that any reward function $r : S \to [0, \bar{r}]$ can be extended to a uniformly-continuous function $F(r) : \mathcal{X} \to \mathcal{A}$ as follows.

$$F(r)(a) = \max\left(0, \max_{a' \in S} r(a') - d(a, a')\frac{\bar{r}}{\eta}\right), \quad a \in \mathcal{A}.$$

Note that this function is $\frac{\bar{r}}{\eta}-$Lipschitz, hence uniformly-continuous—in the case where rewards are stochastic, we can still apply this transformation at the realization level. Further, the sets $B_d(a', \eta)$ for $a' \in S$ are all disjoint by triangular inequality. Thus, for all $a' \in S$, we have $F(r)(a') = r(a')$. We now describe the reduction from uniformly-continuous rewards on $\mathcal{A}$ to unrestricted rewards on S. Let $\mathbb{X} \in$ SOCB$(\mathcal{A})$ and we denote by $\hat{a}_t$ the action selected at time t by a universally consistent learner $f_\cdot$ under $\mathbb{X}$ for uniformly-continuous rewards on $\mathcal{A}$. We now construct a learning rule for unrestricted rewards on S. First, for $a \in \mathcal{A}$, denote by $NN_S(a) = \arg\min_{a' \in S} d(a, a')$ the index of the nearest neighbor of a in S where ties are broken arbitrarily, e.g., by lexicographic order (necessarily, S is countable because $\mathcal{A}$ is separable). We consider the learning rule which selects the actions $NN_S(\hat{a}_t)$, i.e.,

$$f_t^S(\boldsymbol{x}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}, x_t) = NN_S(f_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}, x_t))$$

for all $x_{\leq t} \in \mathcal{X}^t$ and $r_{\leq t-1} \in [0, \bar{r}]^{t-1}$. We aim to show that $f_\cdot^S$ is universally consistent under $\mathbb{X}$ for unrestricted rewards on S. Fix any reward mechanism r on the action space S. We consider the reward mechanism $\tilde{r}$ on the action space $\mathcal{A}$ as follows,

$$\tilde{r}_t(a, x) = F(r(\cdot \mid x))(a),$$

for any $a \in \mathcal{A}$. Note that the mechanism $\tilde{r}$ only depends on the nearest neighbor of selected actions. Denote $\tilde{a}_t$ the corresponding selected action. Observe that by construction of the

functional F, for any $t \geq 1$, $\tilde{r}_t(\tilde{a}_t) \geq \tilde{r}_t(\hat{a}_t)$. Thus, by monotonicity, $f^S_\cdot$ is also consistent on reward mechanism $\tilde{r}$. Now note that $\tilde{f}_\cdot$ only selects actions within S and receives the same rewards that would have been observed by running the learning rule on reward mechanism r. As a result, $f^S_\cdot$ is also consistent for reward r. This ends the proof that it is universally consistent under $\mathbb{X}$ and hence $\mathbb{X} \in \textsc{Socb}(S)$. This ends the proof of the lemma. ■

As a direct consequence of Lemma 5.5 and the results from previous sections, we can use the necessary conditions from the unrestricted reward setting by changing the terms "finite action set" (resp. "countably infinite action set") into "totally-bounded action set" (resp. "non-totally-bounded action set").

Corollary 5.2. *Let $\mathcal{A}$ be a non-totally-bounded metric space. Then, Socb-Uc $\subset$ CS. Let $\mathcal{A}$ be a totally-bounded metric space with $|\mathcal{A}| > 2$. Then, Socb-Uc $\subset$ SMV.*

We now turn to sufficient conditions and show that we can recover the results from the unrestricted case as well. For non-totally-bounded value spaces, the EXPINF learning rule from Theorem 5.11 is already universally consistent under CS processes, which is a necessary condition by Corollary 5.2. As a result, imposing the uniformly-continuous assumption on the rewards does not improve the set of learnable processes.

Theorem 5.12. *Let $\mathcal{X}$ be a separable Borel metrizable space and $\mathcal{A}$ a non-totally-bounded metric space. Then, Socb-Uc = CS.*

Next, we consider totally-bounded action spaces and generalize the learning rule for stochastic rewards in finite action spaces. Recall that this learning rule associates to each time a category $p = \textsc{Category}(t)$, based on the number of previous occurrences of X_t, and works separately on each category. Within each category, the algorithm balances between two strategies: strategy 0 which uses independent EXP3 learners for each distinct instance, and strategy 1 which performs EXPINF. We adapt the algorithm in the following way. First, the EXP3 learners from strategy 0 search for the best action within $\mathcal{A}(\delta_p)$, an δ_p-net of $\mathcal{A}$ where δ_p will be defined carefully. Note that since $\mathcal{A}$ is possibly infinite, restricting strategy 0 to finite action sets is necessary. However, we aim for arbitrary precision, hence we will have $\delta_p \to 0$ as $p \to \infty$. Second, for strategy 1, we use the countable set of functions Π defined as for the EXPINF algorithm in Theorem 5.11.

Theorem 5.13. *Let $\mathcal{A}$ be a totally-bounded metric space. Then, there exists an optimistically universal learning rule for uniformly-continuous rewards, and learnable processes are Socb-Uc = SMV.*

Proof We first define the new learning rule. Category and AssignPurpose are left unchanged. We will use the countable set of policies $\Pi = \{\pi^l, l \geq 1\}$ as in the continuous case in Lemma 5.4, for $\textsc{Explore}(1;\cdot)$, and Algorithm 5.5. Further, in $\textsc{Explore}(0;\cdot)$ and Algorithm 5.5, $\text{EXP3}_{\mathcal{A}}$ is replaced by $\text{EXP3}_{\mathcal{A}(\delta_p)}$. Finally, in SelectStrategy, $\eta_p = 10\frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{2^{p/4}}$ is replaced by $\eta_p = 10\frac{\sqrt{|\mathcal{A}(\delta_p)| \ln |\mathcal{A}(\delta_p)|}}{2^{p/4}}$, where we will define δ_p shortly. In the original proof of the universal consistency of the algorithm, we showed that the average error of the learning rule on category p, $\mathcal{T}_p$ is $\mathcal{O}(\tilde{\epsilon}_p)$ where $\tilde{\epsilon}_p = 2\frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{2^{p/4}}$. Similarly, we now define

$\epsilon_p := 2\frac{\sqrt{|\mathcal{A}(\delta_p)|\ln|\mathcal{A}(\delta_p)|}}{2^{p/4}}$. A key feature of the proof is that since we had $\sum_p \tilde{\epsilon}_p < \infty$, the learner can afford to converge on each set $\mathcal{T}_p$ separately. We mimic this behavior by choosing δ_p such that $\sum_p \epsilon_p < \infty$. Precisely, we pose

$$\delta_p = \min\{2^{-i} : |\mathcal{A}(2^{-i})|\ln|\mathcal{A}(2^{-i})| \leq 2^{p/4}\}.$$

As a result, we obtain directly $\epsilon_p \leq 2^{-1-p/8}$ which is summable, and $\delta_p \to 0$ as $p \to \infty$.

We now show that this learning rule is universally consistent under processes $\mathbb{X} \in \mathrm{SMV}$ by adapting the proof of Theorem 5.8. Fix r a reward mechanism. For every $\epsilon > 0$, there exists $\Delta(\epsilon)$ such that

$$\forall x \in \mathcal{X}, \forall a, a' \in \mathcal{A}, \quad d(a,a') \leq \Delta(\epsilon) \Rightarrow |\bar{r}(a,x) - \bar{r}(a',x)| \leq \epsilon.$$

For every $\delta > 0$, we will also define $\epsilon(\delta) = 2\inf\{\epsilon > 0 : \Delta(\epsilon) \geq \delta\}$. By uniform-continuity, $\epsilon(\delta) \to 0$ as $\delta \to 0$ and because of the factor 2, we have

$$\forall x \in \mathcal{X}, \forall a, a' \in \mathcal{A}, \quad d(a,a') \leq \delta \Rightarrow |\bar{r}(a,x) - \bar{r}(a',x)| \leq \epsilon(\delta).$$

Now observe that in the original proof, the probabilistic bounds $p_i(p,q)$ for $1 \leq i \leq 8$ do not depend on the cardinality of the action set. Therefore, on the same event $\mathcal{E} \cap \mathcal{F}$ of probability one, Eq (5.1), (5.2), (5.3), (5.4), (5.5) and (5.6) hold starting from some time $\hat{T}$, for the intended values of p, q, T. The only difference, however, is that in strategy 0, we perform EXP3 over the restricted action set $\mathcal{A}(\delta_p)$. As a result, for any $x \in \mathcal{X}$, we have

$$\max_{a \in \mathcal{A}(\delta_p)} \bar{r}(a,x) \geq \max_{a \in \mathcal{A}} \bar{r}(a,x) - \epsilon(\delta_p).$$

As a result, Eq (5.1) should be replaced with

$$\hat{R}_p^0(q) \geq \bar{R}_p^*(q) - (T_p^{q+1})^{\frac{7}{8}} - 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1} - T_p^q) - \epsilon(\delta_p)|\mathcal{T}_p(q)|$$

$$\hat{R}_p^0(q) \leq \bar{R}_p^*(q) - (T_p^{q+1})^{\frac{7}{8}} - 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1} - T_p^q).$$

Note that the additional term $\epsilon(\delta_p)|\mathcal{T}_p(q)|$ is not present in the upper bound because searching over $\mathcal{A}$ (in $\bar{R}_p^*(q)$) is always better than searching over $\mathcal{A}(\delta_p)$ (in $\hat{R}_p^0(q)$). Similarly, Eq (5.2) should be replaced with

$$\tilde{R}_p^0(q) \geq \bar{R}_p^*(q) - 6\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{2^{p/4}}(T_p^{q+1} - T_p^q) - (T_p^{q+1})^{3/4} - A_p(q) - \epsilon(\delta_p)|\mathcal{T}_p(q)|.$$

Similarly, the adapted Eq (5.7) becomes

$$\mathcal{R}_p(T) \geq \bar{R}_p^*(T) - \frac{1+c}{2}\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - \frac{T}{2^p} - \epsilon(\delta_p)|\mathcal{T}_p \cap \{t \leq T\}|.$$

Furthering the same bounds, Eq (5.8) becomes

$$\mathcal{R}_p(T) \geq \bar{R}_p^*(T) - (33+5c)\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}T^{1-1/2^7}\log_2 T - 15\epsilon_p T - 2\epsilon(\delta_p)|\mathcal{T}_p \cap \{t \leq T\}|.$$

We are now ready to prove the universal consistency of our learning rule. Fix $0 < \epsilon < 1$, and as in the original proof, let p_0 such that $\sum_{p \geq p_0} \epsilon_p < \frac{\epsilon}{15}$, because $\sum_p \epsilon_p < \infty$. Again, we have $\mathbb{X}^{\leq 4^{p_0}} \in \mathrm{CS}$ and as a result, we can apply Lemma 5.4. As a result, on an event $\mathcal{H}$ of probability one, for all $\epsilon > 0$, there exists $i(\epsilon) \geq 1$ such that $2^{-i(\epsilon)} \leq \Delta(\epsilon), \epsilon$ and $\pi^{i(\epsilon)} \in \Pi$ such that

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^i(X_t)) \\
&\quad \leq \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \mathbb{1}[d(\pi^*(X_t), \pi^{i(\epsilon)}(X_t) \geq 2^{-i(\epsilon)}] \\
&\quad + \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} (\bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^{i(\epsilon)}(X_t)))\mathbb{1}[d(\pi^*(X_t), \pi^{i(\epsilon)}(X_t) \leq \Delta(\epsilon)] \\
&\qquad \leq 2^{-i(\epsilon)} + \epsilon \leq 2\epsilon,
\end{aligned}$$

where π^* denotes the optimal policy. We define the events $\mathcal{E}, \mathcal{F}$ as in the original proof. In the rest of the proof, we will now suppose that the event $\mathcal{E} \cap \mathcal{F} \cap \mathcal{H}$ of probability one is satisfied. On this event, because the parameter $\epsilon > 0$ was arbitrary in the above derivations, there exists $l_0 \geq 1$ (random index) such that

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 4^{p_0}}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t)) \leq \frac{\epsilon}{2^{2p_0+2}}.$$

Following the same arguments as in the original proof, for $p < p_0$, and T_p^q sufficiently large, we need to adapt the following estimates.

$$\begin{aligned}
\max_{1\leq l\leq k(q)} \hat{R}_p^k(q) &\geq \hat{R}_p^{l_0}(q) \\
&\geq \bar{R}_p^{l_0}(q) - (T_p^{q+1})^{7/8} \\
&\geq \bar{R}_p^*(q) - (T_p^{q+1})^{7/8} - \sum_{t\in\mathcal{T}_p(q)} (r_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t))) \\
&\geq \hat{R}_p^0(q) - 2(T_p^{q+1})^{7/8} - 3\epsilon_p(T_p^{q+1} - T_p^q) - \sum_{t\in\mathcal{T}_p(q)} r_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t)).
\end{aligned}$$

Then, observe that

$$\limsup_{q\to\infty} \frac{2(T_p^{q+1})^{7/8} + 3\epsilon_p(T_p^{q+1} - T_p^q) + \sum_{t\in\mathcal{T}_p(q)} r_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t))}{T_p^{q+1} - T_p^q} \leq 4\epsilon_p < \eta_p.$$

Thus, as in the original proof, starting from some time $\tilde{T}$, the learning rule always chooses strategy 1 over strategy 0 for all categories $p \leq p_0$.

We continue the same arguments to obtain for $p < p_0$ and $T \geq 2^{p_0}\tilde{T}$,

$$\mathcal{R}_p(T) - \bar{R}_p^*(T) \geq -2^{p_0}\tilde{T} - 16(3+c)T^{15/16}\ln T - \sum_{t\leq T, t\in\mathcal{T}_p} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t)),$$

which yields

$$\sum_{p<p_0} \bar{R}^*_p(T) - \mathcal{R}_p(T) \leq p_0 2^{p_0} \tilde{T} + 16 p_0 (3+c) T^{15/16} \ln T + \sum_{t \leq T} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t)).$$

Noting that $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^{l_0}(X_t)) \leq \epsilon$, from there, the same arguments show that the learning rule is universally consistent. ■

To summarize, with the uniform-continuity assumption we have generalized all results from the unrestricted rewards case, with a corresponding dichotomy on $\mathcal{A}$ of whether it is totally-bounded or non-totally-bounded.

5.8 Unbounded Rewards

In this last section, we allow for unbounded rewards $\mathcal{R} = [0, \infty)$ and start with the unrestricted rewards setting—no continuity assumption. Recall that in this setting, we assume that for any context $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, the random variable $r(a, x)$ is integrable so that the immediate expected reward is well defined.

When $\mathcal{A}$ is uncountable, we showed that even for bounded rewards, no process $\mathbb{X}$ admits universal learning. Therefore, we will focus on the case when $\mathcal{A}$ is finite or countably infinite, and show that FS determines whether universal consistency is possible. Moreover, a simple variant of EXPINF suffices for optimistically universal learning as follows. Enumerate $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ (or $\mathcal{A} = \{a_1, a_2, \ldots\}$ for countably infinite $\mathcal{A}$) and for any observed instance $x \in \mathcal{X}$, we run an independent EXPINF where the experts of the sequence are the constant policies equal to a_i for $1 \leq i \leq |\mathcal{A}|$, i.e., the expert E_i always selects action a_i.

Theorem 5.14. *Let $\mathcal{A}$ be a countable action set with $|\mathcal{A}| \geq 2$. Then, there is an optimistically universal learning rule and the set of learnable processes admitting universal is FS.*

The fact that FS characterizes universal learning was already the case in the noiseless full-feedback setting as we saw in Chapter 3, hence Theorem 5.14 shows that for unrestricted rewards, we can achieve universal learning in the partial feedback setting without generalization cost.

Proof First, even in the full-information feedback setting, we showed in Chapter 3 that $\mathbb{X} \in \text{FS}$ is necessary for universal consistency. A fortiori in the bandit setting, this condition is still necessary $\textsc{Socb} \subset \text{FS}$.

We now show that the learning rule defined above is universally consistent under FS processes. For simplicity, we denote by $\hat{a}_t$ the action selected by the learning rule at time t. Fix $\mathbb{X} \in \text{FS}$ and define $S = \{x \in \mathcal{X} : \mathbb{X} \cap \{x\} \neq \emptyset\}$ the support of the process. By definition of FS, almost surely, $|S| < \infty$. We denote by $\mathcal{E}$ this event of probability one. Next, for any $x \in S$, we define $\mathcal{T}(x) = \{t : X_t = x\}$ and let $\tilde{S} = \{x \in S : |\mathcal{T}(x)| = \infty\}$ the set of points which are visited an infinite number of times. Recall that the learning rule performs an independent EXPINF subroutine on the times $\mathcal{T}(x)$ for all $x \in S$. As a result, by Corollary 5.1, for any $x \in \tilde{S}$, with probability one, for all $a \in \mathcal{A}$,

$$\limsup_{T\to\infty} \frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \in \mathcal{T}(x), t \leq T} r_t(a) - r_t(\hat{a}_t) \leq 0.$$

Now observe that $\tilde{S}$ is countable. Hence, by the union bound, on an event $\mathcal{F}$ of probability one, for all $x \in \tilde{S}$ and $a \in \mathcal{A}$, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}(x)} r_t(a) - r_t(\hat{a}_t) \leq \limsup_{T\to\infty} \frac{1}{|\mathcal{T}(x)\cap\{t\leq T\}|} \sum_{t\leq T, t\in\mathcal{T}(x)} r_t(a) - r_t(\hat{a}_t) \leq 0.$$

In the rest of the proof, we suppose that $\mathcal{E}\cap\mathcal{F}$ is met. On $\mathcal{E}$, there exists $\hat{T} = 1 + \max\{t : X_t = x, x \in S\setminus\tilde{S}\}$ such that for any $T \geq \hat{T}$, we have $X_t \in \tilde{S}$. Then, for any policy $\pi^* : \mathcal{X}\to\mathcal{A}$, and $T \geq 1$, we have

$$\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq \sum_{t\leq\hat{T}} r_t(\pi^*(X_t)) + \sum_{x\in\tilde{S}} \sum_{t\leq T, t\in\mathcal{T}(x)} r_t(a) - r_t(\hat{a}_t).$$

As a result, because $\mathcal{F}$ is met,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq \sum_{x\in\tilde{S}} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}(x)} r_t(a) - r_t(\hat{a}_t) \leq 0.$$

using the fact that $\mathbb{P}[\mathcal{E}\cap\mathcal{F}] = 1$, we proved that the learning rule is universally consistent under any FS process. This ends the proof of the theorem. ■

The last remaining question is whether this very restrictive set of processes FS can be improved under the continuity and uniform-continuity assumptions from Definition 5.2.

Unfortunately, we show that this is not the case for continuous rewards, however, the continuity assumption allows us to achieve universal consistency on FS processes even on uncountable action spaces. Recall that by Theorem 5.10, universal consistency was not achievable for uncountable spaces in the unrestricted reward case.

Theorem 5.15. *Let $\mathcal{X}$ be a separable metrizable Borel space and $(\mathcal{A}, d)$ be a separable metric space with $|\mathcal{A}| \geq 2$. Then, there is an optimistically universal learning rule for continuous unbounded rewards and the set of learnable processes for universal learning with continuous unbounded rewards is FS.*

Proof In the case of countable action set $\mathcal{A}$ with $|\mathcal{A}| \geq 2$, Theorem 5.14 already showed that FS is sufficient for universal learning under continuous unbounded rewards. Therefore, it remains to show that in the case of uncountable action space, FS is still sufficient for universal learning. More precisely, we will show that the same learning rule which assigns a distinct EXPINF learner to each distinct instance of $\mathbb{X}$ as defined in Theorem 5.14 is still universally consistent under FS processes. The only difference is that we run the learners EXPINF on a dense sequence of actions $(a_i)_{i\geq 1}$ of the complete action set $\mathcal{A}$ which may be uncountable. Let $\mathbb{X} \in$ FS. We use the same notations as in the original proof of Theorem 5.14 for the support $S = \{x \in \mathcal{X} : \mathbb{X}\cap\{x\} \neq \emptyset\}$, the event $\mathcal{E} = \{|S| < \infty\}$, $\mathcal{T}(x) = \{t : X_t = x\}$ for $x \in S$ and $\tilde{S} = \{x \in S : |\mathcal{T}(x)| = \infty\}$. By Corollary 5.1, for any $x \in \tilde{S}$, with probability one, for all $i \geq 1$, we have now

$$\limsup_{T\to\infty} \frac{1}{|\mathcal{T}(x)\cap\{t\leq T\}|} \sum_{t\in\mathcal{T}(x), t\leq T} r_t(a_i) - r_t(\hat{a}_t) \leq 0.$$

Let $a \in \mathcal{A}$ and $\epsilon > 0$, because $(a_i)_{i \geq 1}$ is dense in $\mathcal{A}$ and the immediate reward is continuous, there exists $i(\epsilon)$ such that $|\bar{r}(a_{i(\epsilon)}) - \bar{r}(a)| \leq \epsilon$. Now observe that by the union bound, for any $x \in \tilde{S}$, with probability one, by the law of large numbers, one has for all $i \geq 1$,

$$\frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \in \mathcal{T}(x), t \leq T} r_t(a_i) \underset{T \to \infty}{\longrightarrow} \bar{r}_t(a_i),$$

and similarly for a. As a result, for any $x \in \tilde{S}$, with probability one, for any $\epsilon > 0$,

$$\begin{aligned}
&\limsup_{T \to \infty} \frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \in \mathcal{T}(x), t \leq T} r_t(a) - r_t(\hat{a}_t) \\
&\leq \bar{r}(a) - \bar{r}(a_{i(\epsilon)}) + \limsup_{T \to \infty} \frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \in \mathcal{T}(x), t \leq T} r_t(a_{i(\epsilon)}) - r_t(\hat{a}_t) \\
&\leq \epsilon.
\end{aligned}$$

As a result, we showed that for any $x \in \tilde{S}$, and any $a \in \mathcal{A}$, with probability one,

$$\limsup_{T \to \infty} \frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \in \mathcal{T}(x), t \leq T} r_t(a) - r_t(\hat{a}_t) \leq 0.$$

Now fix $\pi^* : \mathcal{X} \to \mathcal{A}$ a measurable policy. Because $\tilde{S}$ is countable, by the union bound, on an event $\mathcal{F}$ of probability one, for all $x \in \tilde{S}$, we have

$$\begin{aligned}
&\limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}(x)} r_t(\pi^*(x)) - r_t(\hat{a}_t) \\
&\leq \limsup_{T \to \infty} \frac{1}{|\mathcal{T}(x) \cap \{t \leq T\}|} \sum_{t \leq T, t \in \mathcal{T}(x)} r_t(\pi^*(x)) - r_t(\hat{a}_t) \leq 0.
\end{aligned}$$

Then, the same arguments as in the original proof show that on $\mathcal{E} \cap \mathcal{F}$, for any $T \geq 1$, one has

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq \sum_{x \in \tilde{S}} \limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}(x)} r_t(a) - r_t(\hat{a}_t) \leq 0.$$

Thus, the learning rule is universally consistent under FS processes.

We now show that $\mathbb{X} \in \text{FS}$ is still necessary for universal learning with continuous rewards. For the unrestricted reward case, this is a direct consequence of our results in Section 3.9 from Chapter 3, which we now adapt for continuous rewards. First, for any $\mathbb{X} \notin \text{FS}$, we showed there that there exists a disjoint measurable partition $\{B_i\}_{i=1}^{\infty}$ such that with non-zero probability, $|\{i : \mathbb{X} \cap B_i \neq \emptyset\}| = \infty$ on an event $\mathcal{E}_0$. Then, we constructed a sequence of times T_i for $i \geq 1$ such that on an event $\mathcal{E}$ of probability one, for sufficiently large indices i, $\tau_i := \min\{0\} \cup \{t : X_t \in B_i\} \leq T_i$. Now fix two distinct actions $a_0, a_1 \in \mathcal{A}$, let $\epsilon = \frac{d(a_0, a_1)}{3}$ and fix a learning rule $f.$. We denote by $\hat{a}_t$ its selected action at time t. Consider the following rewards

$$r^{\boldsymbol{U}}(a, x) = \max\left(0, T_i\left(1 - \frac{d(a, a_{U_j})}{\epsilon}\right)\right), \quad x \in B_j, \tag{5.11}$$

for any binary sequence $\boldsymbol{U}$. Now suppose that they were sampled from an i.i.d. sequence of Bernoullis $\mathcal{B}(\frac{1}{2})$, independent of the process $\mathbb{X}$ and the randomness of the learning rule. Now observe that for any $i \geq 1$ such that $\tau_i \leq T_i$, with probability at least $\frac{1}{2}$ independently of the past, we have $\hat{a}_{\tau_i} \notin B(a_{U_j}, \epsilon)$, which implies $\max_{a \in \mathcal{A}} r^{\boldsymbol{U}}_{\tau_i}(a) - r^{\boldsymbol{U}}_{\tau_i}(\hat{a}_{\tau_i}) \geq T_i$. From there, the same arguments as in the original proof show that with probability one, this event occurs infinitely often and $\mathcal{E}$ is met, which by the law of total probability implies that there exists a deterministic choice of values for $\boldsymbol{U} = (U_j)_{j \geq 1}$ such that on the corresponding deterministic (hence stationary) rewards, the learning rule is not consistent on $\mathcal{E}_0 \cap \mathcal{E}$ which has non-zero probability. This shows that $\mathbb{X}$ does not admit universal learning even in the simplest case of deterministic continuous rewards. ■

Last, we investigate the case of uniformly-continuous unrestricted rewards. Unfortunately, the uniform continuity assumption over the immediate expected rewards does not provide any advantage over the continuity assumption.

Proposition 5.2. *Let $\mathcal{X}$ be a separable metrizable Borel space and $\mathcal{A}$ be a separable metric space with $|\mathcal{A}| \geq 2$. Then, the set of learnable processes for universal learning with uniformly-continuous unbounded rewards is FS.*

Proof It suffices to show that the FS condition is still necessary for universal learning under uniformly-continuous rewards since the sufficiency is guaranteed by Theorem 5.15. We adapt the proof of the necessity of FS in the continuous unbounded reward case. Let $\mathbb{X} \notin \text{FS}$ and suppose that there exists a universally consistent learning rule f. under $\mathbb{X}$ for uniformly-continuous unbounded rewards. We use the same notations as in the proof of Theorem 5.15. We now define a sequence $(M_i)_{i \geq 1}$ recursively such that $M_1 = 2T_1$ and for any $i \geq 1$, $M_{i+1} = 2T_{i+1} + 4T_{i+1} \sum_{j \leq i} M_j$. Then, consider the following stochastic rewards

$$r(a, x) = \begin{cases} M_i \left(1 + \frac{d(a,a_0) \wedge d(a_0,a_1)}{d(a_0,a_1)}\right) & \text{w.p. } \frac{1}{2}, \\ M_i \left(1 - \frac{d(a,a_0) \wedge d(a_0,a_1)}{d(a_0,a_1)}\right) & \text{w.p. } \frac{1}{2}. \end{cases} \quad x \in B_i, i \geq 1.$$

These rewards are uniformly-continuous because for any $x \in \mathcal{X}$, the expected immediate reward is $\bar{r}(a, x) = 0$ for all $a \in \mathcal{A}$. Now for $u \in \{0, 1\}$, define the constant policy $\pi^u : x \in \mathcal{X} \mapsto a_u \in \mathcal{A}$. Denote by $\hat{a}_t$ the action selected by the learning rule at time t. Because it is consistent under the rewards mechanism given by r, using π^0, π^1 and the union bound, we have that almost surely, for any $u \in \{0, 1\}$,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(a_u, X_t) - r_t(\hat{a}_t, X_t) \leq 0. \tag{5.12}$$

Now recall that on the event $\mathcal{E}_0$ of non-zero probability, we have $|\{i : \mathbb{X} \cap B_i \neq \emptyset\}| = \infty$. In other terms, $|\{i : \tau_i > 0\}| = \infty$. We then define the random sequence of indices $(i_k)_{k \geq 1}$ such that on $\mathcal{E}_0^c$, $i_k = 0$ for all $k \geq 1$ and on $\mathcal{E}_0$, the indices are defined recursively such that $i_1 = \arg\min_{i \geq 1, \tau_i > 0} \tau_i$ and for $k \geq 1$, we have $i_{k+1} = \arg\min_{i > i_k, \tau_i > 0} \tau_i$. The arg min are well defined because on $\mathcal{E}_0$, all the times τ_i for $i \in \{j \geq 1 : \tau_j > 0\}$ are distinct. As a result,

by the construction of the recursion, on $\mathcal{E}_0$, the sequence $(i_k)_{k\geq 1}$ is an increasing sequence of times and for all $k \geq 1$, we have

$$\{i : \mathbb{X}_{<\tau_{i_k}} \cap B_i \neq \emptyset\} = \{i : 0 < \tau_i < \tau_{i_k}\} \subset \{1 \leq i < i_k\}.$$

Now recall that on the event $\mathcal{E}$ of probability one, there exists $\hat{i} \geq 1$ such that for any $i \geq \hat{i}$, we have $\tau_i := \min\{0\} \cup \{t : X_t \in B_i\} \leq T_i$. Therefore, on $\mathcal{E}_0 \cap \mathcal{E}$, letting $\hat{k} = \min\{k : i_k \geq \hat{i}\}$, we have that for $k \geq \hat{k}$, and $u \in \{0, 1\}$

$$\begin{aligned}
\sum_{t=1}^{\tau_{i_k}-1} r_t(a_u, X_t) - r_t(\hat{a}_t, X_t) &\geq \sum_{i:\mathbb{X}_{<\tau_{i_k}} \cap B_i \neq \emptyset} \sum_{t<\tau_{i_k}, X_t \in B_i} (-2M_i) \\
&\geq -2 \sum_{i<i_k} T_{i_k} M_i \\
&\geq -\frac{M_{i_k}}{2} + T_{i_k}.
\end{aligned}$$

Now observe that on the event $\mathcal{E}_0 \cap \mathcal{E}$ which has non-zero probability, if $d(\hat{a}_{\tau_{i_k}}, a_0) \geq \frac{d(a_0,a_1)}{2}$ and the reward on B_{i_k} at time τ_{i_k} is in its negative alternative, i.e., $r(a, x) = M_i \left(1 - \frac{d(a,a_0) \wedge d(a_0,a_1)}{d(a_0,a_1)}\right)$, we have

$$\frac{1}{\tau_{i_k}} \sum_{t=1}^{\tau_{i_k}} r_t(a_0, X_t) - r_t(\hat{a}_t, X_t) \geq \frac{1}{\tau_{i_k}} \left(\frac{M_{i_k}}{2} - \frac{M_{i_k}}{2} + T_{i_k} \right) \geq 1.$$

Now by construction, the negative alternative occurs with probability $\frac{1}{2}$, independently from the past history and the complete process $\mathbb{X}$. As a result, for any $k \geq 1$, we have

$$\mathbb{P}\left[\frac{1}{\tau_{i_k}} \sum_{t=1}^{\tau_{i_k}} r_t(a_0, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}, d(\hat{a}_{\tau_{i_k}}, a_0) \geq \frac{d(a_0, a_1)}{2} \right] \geq \frac{1}{2}. \tag{5.13}$$

Similarly, one can check that on the event $\mathcal{E}_0 \cap \mathcal{E}$, if $d(\hat{a}_{\tau_{i_k}}, a_0) < \frac{d(a_0,a_1)}{2}$ and the reward on B_{i_k} at time τ_{i_k} is in its positive alternative, we have

$$\frac{1}{\tau_{i_k}} \sum_{t=1}^{\tau_{i_k}} r_t(a_1, X_t) - r_t(\hat{a}_t, X_t) \geq \frac{1}{\tau_{i_k}} \left(\frac{M_i}{2} - \frac{M_{i_k}}{2} + T_{i_k} \right) \geq 1.$$

As a result, the same arguments as above give

$$\mathbb{P}\left[\frac{1}{\tau_{i_k}} \sum_{t=1}^{\tau_{i_k}} r_t(a_1, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}, d(\hat{a}_{\tau_{i_k}}, a_0) < \frac{d(a_0, a_1)}{2} \right] \geq \frac{1}{2}. \tag{5.14}$$

Finally, define for any $T \geq 1$ the event

$$\mathcal{F}_T = \left\{ \frac{1}{T} \sum_{t=1}^{T} r_t(a_0, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \right\} \cup \left\{ \frac{1}{T} \sum_{t=1}^{T} r_t(a_1, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \right\}.$$

We obtain for any $k \geq 1$,

$$\begin{aligned}
&\mathbb{P}[\mathcal{F}_{\tau_{i_k}} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}] \\
&\geq \mathbb{P}\left[\mathcal{F}_{\tau_{i_k}} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}, d(\hat{a}_{\tau_{i_k}}, a_0) \geq \frac{d(a_0, a_1)}{2}\right] \mathbb{P}\left[d(\hat{a}_{\tau_{i_k}}, a_0) \geq \frac{d(a_0, a_1)}{2} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}\right] \\
&+ \mathbb{P}\left[\mathcal{F}_{\tau_{i_k}} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}, d(\hat{a}_{\tau_{i_k}}, a_0) < \frac{d(a_0, a_1)}{2}\right] \mathbb{P}\left[d(\hat{a}_{\tau_{i_k}}, a_0) < \frac{d(a_0, a_1)}{2} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}\right] \\
&\geq \frac{1}{2}\mathbb{P}\left[d(\hat{a}_{\tau_{i_k}}, a_0) \geq \frac{d(a_0, a_1)}{2} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}\right] + \frac{1}{2}\mathbb{P}\left[d(\hat{a}_{\tau_{i_k}}, a_0) < \frac{d(a_0, a_1)}{2} \mid \mathcal{E}_0, \mathcal{E}, k \geq \hat{k}\right] \\
&= \frac{1}{2},
\end{aligned}$$

where in the second inequality we used Eq (5.13) and Eq (5.14). As a result, using Fatou's lemma

$$\begin{aligned}
\mathbb{P}[\mathcal{F}_{\tau_{i_k}} \text{occurs for infinitely many } k \geq 1 \mid \mathcal{E}_0, \mathcal{E}] &\geq \limsup_{k \geq 1} \mathbb{P}[\mathcal{F}_{\tau_{i_k}} \mid \mathcal{E}_0, \mathcal{E}] \\
&\geq \frac{1}{2} \limsup_{k \geq 1} \mathbb{P}[k \geq \hat{k} \mid \mathcal{E}_0, \mathcal{E}] = \frac{1}{2},
\end{aligned}$$

where in the last inequality, we used the dominated convergence theorem given that on the event $\mathcal{E}$, $\hat{k} < \infty$. As a result, we showed that

$$\mathbb{P}\left[\exists u \in \{0,1\}, \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(a_u, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \mid \mathcal{E}_0, \mathcal{E}\right] \geq \frac{1}{2}.$$

However, because $\mathbb{P}[\mathcal{E} \cap \mathcal{E}_0] = \mathbb{P}[\mathcal{E}_0] > 0$, Eq (5.12) shows that

$$\mathbb{P}\left[\forall u \in \{0,1\}, \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(a_u, X_t) - r_t(\hat{a}_t, X_t) \geq 1 \mid \mathcal{E}_0, \mathcal{E}\right] = 1,$$

which contradicts the previous inequality. This shows that the learning rule was not consistent under the rewards $(r_t)_t$, hence not universally consistent under $\mathbb{X}$. This shows that FS is necessary for universal learning and completes the proof. ■

5.9 Appendix

We first give the proofs of the characterizations for the classes of stochastic processes CS and SMV.

5.9.1 Proof of Lemma 5.1

Suppose $\mathbb{X} \notin$ CS. By [Han21a, Lemma 14], there exists a disjoint sequence $\{B_i\}_{i=1}^{\infty}$ of measurable subsets of $\mathcal{X}$ such that, on an event $\mathcal{E}_0$ of probability strictly great than 0, it

holds that

$$\lim_{j\to\infty} \hat{\mu}_{\mathbb{X}}\left(\bigcup_{i\geq j} B_i\right) > 0.$$

Without loss of generality, we may suppose $B_1 = \mathcal{X} \setminus \bigcup_{i>1} B_i$ so that $\bigcup_{i\in\mathbb{N}} B_i = \mathcal{X}$. Define a random variable α as

$$\alpha = \lim_{j\to\infty} \hat{\mu}_{\mathbb{X}}\left(\bigcup_{i\geq j} B_i\right).$$

Inductively define sequences T_k, J_k in $\mathbb{N}$ as follows. Let $T_0 = 0$ and $J_0 = 1$. For each $k \in \mathbb{N}$, suppose T_{k-1} and J_{k-1} are defined, elements of $\mathbb{N}$, and define T_k and J_k as follows. Note that, by definition of $\hat{\mu}_{\mathbb{X}}$, there exists an $\mathbb{X}$-dependent random variable $\tau_k \in \mathbb{N}$ with $\tau_k > T_{k-1}$ such that

$$\frac{1}{\tau_k}\left|\mathbb{X}_{\leq\tau_k} \cap \bigcup_{i\geq J_{k-1}} B_i\right| \geq (1/2)\hat{\mu}_{\mathbb{X}}\left(\bigcup_{i\geq J_{k-1}} B_i\right).$$

Moreover, by monotonicity of $\hat{\mu}_{\mathbb{X}}(\cdot)$, the right-hand side is no smaller than $\alpha/2$. Let $T_k \in \mathbb{N}$ be any finite non-random value such that

$$\mathbb{P}(\tau_k > T_k) < \mathbb{P}(\mathcal{E}_0)2^{-k-2}.$$

Next note that, since the sets B_i are disjoint, there exists a finite $\mathbb{X}$-dependent random variable $j_k \in \mathbb{N}$ with $j_k > J_{k-1}$ such that

$$\mathbb{X}_{\leq T_k} \cap \bigcup_{i\geq j_k} B_i = \emptyset.$$

Let $J_k \in \mathbb{N}$ be any finite non-random value such that

$$\mathbb{P}(j_k > J_k) < \mathbb{P}(\mathcal{E}_0)2^{-k-2}.$$

In particular, on the event that $j_k \leq J_k$, it holds that

$$\mathbb{X}_{\leq T_k} \cap \bigcup_{i\geq J_k} B_i = \emptyset,$$

which implies that

$$\mathbb{X}_{\leq T_k} \cap \bigcup_{i\geq J_{k-1}} B_i = \mathbb{X}_{\leq T_k} \cap \bigcup_{J_{k-1}\leq i<J_k} B_i.$$

Thus, if both events $\tau_k \leq T_k$ and $j_k \leq J_k$ hold, it must be that

$$\frac{1}{\tau_k}\left|\mathbb{X}_{\leq\tau_k} \cap \bigcup_{J_{k-1}\leq i<J_k} B_i\right| \geq \alpha/2,$$

or equivalently,

$$\frac{1}{\tau_k}\sum_{t=1}^{\tau_k}\mathbb{1}[i_t \in \{J_{k-1} \le i < J_k\}] \ge \alpha/2. \tag{5.15}$$

This completes the inductive definition of the sequences T_k and J_k.

To specify the N_i values, for each $k \in \mathbb{N}$ and $i \in \{J_{k-1}, \dots, J_k - 1\}$, define $N_i = T_k$. Note that the event $\mathcal{E}_1 = \mathcal{E}_0 \cap \bigcap_{k\in\mathbb{N}} \{\tau_k \le T_k\} \cap \{j_k \le J_k\}$ has a probability of at least

$$\mathbb{P}(\mathcal{E}_0) - \sum_{k\in\mathbb{N}} \mathbb{P}(\mathcal{E}_0) 2^{-k-1} = \mathbb{P}(\mathcal{E}_0)/2 > 0$$

by the union bound. On the event $\mathcal{E}_1$, (5.15) holds for every $k \in \mathbb{N}$. Since $\tau_k \le T_k$ on $\mathcal{E}_1$, we also trivially have that every $i \in \{J_{k-1}, \dots, J_k - 1\}$ and $t \in [\tau_k]$ satisfy $|\mathbb{X}_{<t} \cap B_i| < t \le T_k = N_i$. Together, these facts imply that on $\mathcal{E}_1$, every $k \in \mathbb{N}$ satisfies

$$\frac{1}{\tau_k}\sum_{t=1}^{\tau_k}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] \ge \alpha/2.$$

Since we also have $\alpha > 0$ on the event $\mathcal{E}_1$, and since T_k is strictly increasing, and $\tau_k > T_{k-1}$ implies $\tau_k \to \infty$ as $k \to \infty$, altogether we have that on the event $\mathcal{E}_1$,

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] \\
&\ge \limsup_{k\to\infty} \frac{1}{\tau_k}\sum_{t=1}^{\tau_k}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] \ge \alpha/2 > 0.
\end{aligned}$$

We establish the final claim that such a result is not possible for $\mathbb{X} \in \mathrm{CS}$, as follows. Fix any $\mathbb{X} \in \mathrm{CS}$. For any disjoint sequence B_i of measurable subsets of $\mathcal{X}$, and any sequence $N_i \in \mathbb{N}$, define $C_n = \bigcup\{B_i : N_i > n\}$, and note that $C_n \downarrow \emptyset$. For every $n \in \mathbb{N}$, we have

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < N_{i_t}] \\
&\le \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < n] + \mathbb{1}[N_{i_t} > n]) \\
&\le \left(\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < n]\right) + \hat{\mu}_{\mathbb{X}}(C_n)\,.
\end{aligned} \tag{5.16}$$

For any $m \in \mathbb{N}$, any $t \ge m$ has $\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < n] \le \mathbb{1}[|\mathbb{X}_{<m} \cap B_{i_t}| < n]$, so that

$$\begin{aligned}
&\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < n] \\
&\le \limsup_{T\to\infty} \frac{m}{T} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[|\mathbb{X}_{<m} \cap B_{i_t}| < n] = \hat{\mu}_{\mathbb{X}}\Big(\bigcup\{B_i : |\mathbb{X}_{<m} \cap B_i| < n\}\Big)\,.
\end{aligned}$$

Since the first expression above has no dependence on m, the conclusion remains valid in the limit of $m \to \infty$, so that

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[|\mathbb{X}_{<t} \cap B_{i_t}| < n] \leq \lim_{m\to\infty} \hat{\mu}_{\mathbb{X}}\Big(\bigcup\{B_i : |\mathbb{X}_{<m} \cap B_i| < n\}\Big)\,,$$

which equals zero almost surely (by Lemmas 13 and 14 of [Han21a]). Altogether, for any $n \in \mathbb{N}$, with probability one, (5.16) is at most $\hat{\mu}_{\mathbb{X}}(C_n)$. Again, since (5.16) has no dependence on n, this inequality remains valid in the limit as $n \to \infty$, so that with probability one, (5.16) is at most

$$\lim_{n\to\infty} \hat{\mu}_{\mathbb{X}}(C_n)\,,$$

which equals zero almost surely (by Lemma 13 of [Han21a]). The conclusion that (5.16) equals zero almost surely follows by the union bound.

5.9.2 Proof of Proposition 5.1

We start by showing (2) $\Rightarrow$ (1). Suppose that a process $\mathbb{X}$ is not in SMV. We aim to show that $\mathbb{X}$ disproves the second property. Because $\mathbb{X} \notin$ SMV, there exists a sequence of disjoint measurable sets $(B_i)_{i\geq 1}$, $\epsilon, \delta > 0$ such that with probability $\delta > 0$

$$\limsup_{T\to\infty} \frac{|\{i : \mathbb{X}_{\leq T} \cap B_i \neq \emptyset\}|}{T} \geq \epsilon.$$

Denote by $\mathcal{A}$ this event, and consider the sets $A_i = \bigcup_{j\geq i} B_i$ for $i \geq 1$. Now fix $i \geq 1$. For any $T \geq 1$, we have

$$\sum_{t\leq T, t\in\mathcal{T}^{\leq 1}} \mathbb{1}_{A_i}(X_t) = |A_i \cap \mathbb{X}_{\leq T}| \geq |\{j \geq i : B_j \cap \mathbb{X}_{\leq T} \neq \emptyset\}| \geq |\{j : \mathbb{X}_{\leq T} \cap B_j \neq \emptyset\}| - (i-1),$$

where in the first inequality we used the fact that the B_j are disjoint for all $j \geq i$, but included within A_i. As a result, on the event $\mathcal{A}$ we have $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 1}} \mathbb{1}_{A_i}(X_t) \geq \epsilon$. Hence,

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\leq 1}} \mathbb{1}_{A_i}(X_t)\right] \geq \epsilon \mathbb{P}[\mathcal{A}] = \epsilon\delta.$$

This holds for all $i \geq 1$ but $A_i \downarrow \emptyset$, which shows that $\mathbb{X}$ does not satisfy property (2).

To prove (1) $\Rightarrow$ (2), now suppose that property (2) is not satisfied by $\mathbb{X}$. We aim to show that $\mathbb{X} \notin$ SMV. Then, there exists a sequence of measurable sets $A_i \downarrow \emptyset$, $\epsilon > 0$ and an increasing sequence of indices $(i_k)_{k\geq 1}$ such that for all $k \geq 1$

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{|A_{i_k} \cap \mathbb{X}_{\leq T}|}{T}\right] \geq \epsilon.$$

Because the sets A_i are decreasing and the quantity within the expectation is increasing in the set A, this shows that for all $i \geq 1$, we have $\mathbb{E}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T}\right] \geq \epsilon$. Therefore, for

any $i \geq 1$ because $\mathbb{E}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T}\right] \leq \mathbb{P}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}\right] + \frac{\epsilon}{2}$ we obtain for all $i \geq 1$

$$\mathbb{P}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}\right] \geq \frac{\epsilon}{2}.$$

Again, because the inner quantity is increasing in the set A, we obtain

$$\begin{aligned}
\mathbb{P}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}, \forall i \geq 1\right] &= \lim_{I\to\infty} \mathbb{P}\left[\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}, 1 \leq i \leq I\right] \\
&= \lim_{I\to\infty} \mathbb{P}\left[\limsup_{T\to\infty} \frac{|A_I \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}\right] \\
&\geq \frac{\epsilon}{2}.
\end{aligned}$$

We will denote by $\mathcal{H}$ this event in which for all $i \geq 1$, we have $\limsup_{T\to\infty} \frac{|A_i \cap \mathbb{X}_{\leq T}|}{T} \geq \frac{\epsilon}{2}$. Under the event $\mathcal{H}$, for any $i, t^0 \geq 1$, there always exists $t^1 > t^0$ such that $\frac{|A_i \cap \mathbb{X}_{\leq t^1}|}{t^1} \geq \frac{\epsilon}{4}$. We construct a sequence of times $(t_p)_{p\geq 1}$ and indices $(i_p)_{p\geq 1}$, $(u_p)_{p\geq 1}$ by induction as follows. We first pose $i_1 = t_0 = 0$. Now assume that for $p \geq 1$, the time t_{p-1} and index i_p are defined. Let $t_p > t_{p-1}$ such that

$$\mathbb{P}\left[\mathcal{H}^c \cup \bigcup_{t_{p-1}<t\leq t_p} \left\{\frac{|A_{i_p} \cap \mathbb{X}_{\leq t}|}{t} \geq \frac{\epsilon}{4}\right\}\right] \geq 1 - \frac{\epsilon}{2^{p+3}}.$$

This is also possible because $\mathcal{H} \subset \bigcup_{t>t_{p-1}} \left\{\frac{|A_{i_p} \cap \mathbb{X}_{\leq t}|}{t} \geq \frac{\epsilon}{4}\right\}$. Last, let $i_{p+1} > i_p$ such that $\mathbb{P}[A_{i_{p+1}} \cap \mathbb{X}_{\leq t_p} \neq \emptyset] \leq \frac{\epsilon}{2^{p+3}}$ which is possible since $A_u \downarrow \emptyset$ as $u \to \infty$. We denote $\mathcal{E}_p$ this event. Then,

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{H}^c \cup \bigcup_{t_{p-1}<t\leq t_p} \left\{\frac{|(A_{i_p} \setminus A_{i_{p+1}}) \cap \mathbb{X}_{\leq t}|}{t} \geq \frac{\epsilon}{4}\right\}\right] \\
&\geq \mathbb{P}\left[\mathcal{E}_p \cap \mathcal{H}^c \cup \bigcup_{t_{p-1}<t\leq t_p} \left\{\frac{|A_{i_p} \cap \mathbb{X}_{\leq t}|}{t} \geq \frac{\epsilon}{4}\right\}\right] \geq 1 - \frac{\epsilon}{2^{p+2}}.
\end{aligned}$$

We denote $\mathcal{F}_p$ this event. This ends the recursive construction of times t_p and indices i_p for all $p \geq 1$. Note that by construction, $\mathbb{P}[\mathcal{F}_p^c] \leq \frac{\epsilon}{2^{p+2}}$. Hence, by union bound, the event $\mathcal{H} \cap \bigcap_{p\geq 1} \mathcal{F}_p$ has probability $\mathbb{P}[\mathcal{H} \cap \bigcap_{p\geq 1} \mathcal{F}_p] \geq \mathbb{P}[\mathcal{H}] - \frac{\epsilon}{4} \geq \frac{\epsilon}{4}$. For conciseness, denote $B_p = A_{i_p} \setminus A_{i_{p+1}}$. On the event $\mathcal{H} \cap \bigcap_{p\geq 1} \mathcal{F}_p$ we showed that for all $p \geq 1$, there exists $t_{p-1} < t \leq t_p$ such that $|B_p \cap \mathbb{X}_{\leq t}| \geq \frac{\epsilon}{4} t$, and $(B_p)_{p\geq 1}$ is a sequence of disjoint measurable sets.

Now for any $p \geq 1$, we will construct a countable partition of B_p that separates all points falling in B_p within time horizon t_p. Let $\delta_p > 0$ such that

$$\mathbb{P}\left[\min_{u,v\leq t_p : X_u \neq X_v} \rho(X_u, X_v) \leq \delta_p\right] \leq \frac{\epsilon}{2^{p+3}}.$$

We denote by $\mathcal{G}_p$ the complementary of this event. Note that $\mathbb{P}[\bigcup_{p\geq 1}\mathcal{G}_p^c] \leq \frac{\epsilon}{8}$. As a result, the event $\mathcal{I} := \mathcal{H} \cap \bigcap_{p\geq 1}(\mathcal{F}_p \cap \mathcal{G}_p)$ has probability at least $\frac{\epsilon}{8}$. We will show that in this event, $\mathbb{X}$ disproves the SMV condition. Precisely, let $(x^i)_{i\geq 1}$ a dense sequence of $\mathcal{X}$. We will denote the balls of $\mathcal{X}$ by $B(x,r) = \{x' : \rho(x,x') < r\}$. Define the following partition of $\mathcal{X}$,

$$\mathcal{P}(\delta): \quad P_i(\delta) = B(x^i,\delta) \setminus \bigcup_{j<i} B(x^j,\delta).$$

Finally, for any $p, i \geq 1$, define $P_i^p := P_i(\delta_p) \cap B_p$. We can note that $\bigcup_{i\geq 1} P_i^p = B_p$. Further, the sets $(B_i^p)_{i,p\geq 1}$ are all disjoint, and form a countable sequence. However, on the event $\mathcal{I}$, for every $p \geq 1$, there exists a time $t_{p-1} < t \leq t_p$ such that $|B_p \cap \mathbb{X}_{\leq t}| \geq \frac{\epsilon}{4}t$. But because the event $\mathcal{G}_p$ is satisfied, all the points falling in B_p within horizon $t \leq t_p$ are separated by at least δ_p, hence fall in distinct sets B_i^p. As a result,

$$|\{i \geq 1 : P_i^p \cap \mathbb{X}_{\leq t} \neq \emptyset\}| \geq |B_p \cap \mathbb{X}_{\leq t}| \geq \frac{\epsilon}{4}t.$$

This shows that on the event $\mathcal{I}$, for every $p \geq 1$, there exists $t > t_{p-1}$ such that $|\{i, p \geq 1 : P_i^p \cap \mathbb{X}_{\leq t} \neq \emptyset\}| \geq \frac{\epsilon}{4}t$, and as a result

$$\limsup_{T\to\infty} \frac{|\{i, p \geq 1 : P_i^p \cap \mathbb{X}_{\leq T} \neq \emptyset\}|}{T} \geq \frac{\epsilon}{4}.$$

The fact that $\mathbb{P}[\mathcal{I}] \geq \frac{\epsilon}{8}$ ends the proof that $\mathbb{X} \notin$ SMV, and that the first proposition is equivalent to SMV.

We now show the equivalence (2) $\Leftrightarrow$ (3). We clearly have (3) $\Rightarrow$ (2). Now suppose that $\mathbb{X}$ satisfies (2). Let $M > 1$ and A be a measurable set. Then, for any $T \geq 1$, we have

$$\frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}^{\leq M}} \mathbb{1}_A(X_t) \leq M\frac{|A\cap\mathbb{X}_{\leq t}|}{T} = \frac{M}{T}\sum_{t\leq T, t\in\mathcal{T}^{\leq 1}} \mathbb{1}_A(X_t).$$

Because $(X_t)_{t\in\mathcal{T}^{\leq 1}} \in$ CS, we obtain as a result $(X_t)_{t\in\mathcal{T}^{\leq M}} \in$ CS using the definition. This ends the proof of the proposition.

Chapter 6

Adversarial Contextual Bandits

6.1 Introduction

The previous Chapter 5 studied the fundamental question of *learnability* for contextual bandits under the standard *stationarity* assumption for the rewards. This classical assumption in the contextual bandit literature is that the underlying dependency between rewards, contexts and actions of the learner, is invariant over time. In the present chapter, we challenge this assumption and provide some characterizations of universal learning in non-stationary environements. This is captured by the adversarial contextual bandit framework which can be seen as a crucial step towards mode complex non-stationary machine learning frameworks such as reinforcement learning.

The universal learning framework we use in this chapter is the same as the one introduced in Chapters 2 and 5 for contextual bandits. We briefly recall the setup here, and specify the differences between stationary and non-stationary reward environments. The contextual bandit setting is a central problem in statistical decision-making. This setting models the interaction between a learner or decision maker, and a reward mechanism. At each iteration of the learning process, the learner observes a *context* $x \in \mathcal{X}$ (also known as covariate in the statistical learning literature), then selects an *action* $a \in \mathcal{A}$ to perform. The decision maker then receives a reward based on the context and selected action, which can then be used to perform informed future actions. The major difference with the standard supervised learning framework is that the learner can only observe the reward of the selected action, referred to as partial feedback, instead of the full-feedback case of supervised learning in which a learner can directly compute the reward (or loss) of non-selected actions. New phenomena arise from these characteristics, including the well-known exploration/exploitation trade-off: algorithms should balance between exploiting known high-reward actions and exploring new actions that potentially could yield higher rewards.

Universal consistency. We focus on the foundational notion of *consistency*. In the contextual bandit context, a learner is consistent if its long-term excess regret vanishes. Contexts are modeled by a stochastic process $\mathbb{X} = (X_t)_{t\geq 1}$. If $\hat{a}_t$ is the selected action and r_t the reward

function at time t, we ask that for any measurable policy π^*,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq 0 \quad (a.s.).$$

As shown in the above equation, we follow a traditional regret analysis, where we compare the learner to a fixed policy (static regret) as opposed to switching regret where the comparison policy may also change. For generality, one commonly aims to design algorithms that ensure consistency for a large class of instances. We consider the strongest notion of *universal consistency*, introduced in [Han21a], which asks that a learning rule is consistent for any possible reward mechanism for $(r_t)_{t\geq 1}$, within a specified reward model.

A simple example of reward model is the stationary reward model: rewards r_t are given by a time-invariant conditional distribution $P_{r|x,a}$, conditioned on the current context X_t and the selected action $\hat{a}_t$. In this model, a learning rule is universally consistent if for any conditional distribution $P_{r|x,a}$—any stationary rewards—the algorithm is consistent. This was precisely the model studied in Chapter 5. In the present chapter, we go beyond by considering diverse models of non-stationary and adversarial reward mechanisms.

The notion of universal consistency was mostly studied in the full-feedback supervised learning framework in which observes a stream of data $(X_t, Y_t)_{t\geq 1}$, makes predictions $\hat{Y}_t$ at each step, and receives rewards (losses) according to some loss measuring the discrepancy between predictions and true values Y_t. The main difference with contextual bandits is that the knowledge of Y_t allows to compute losses for *any* previous prediction, whereas in contextual bandits, we only receive reward information for the *only* action selected at that time period. This full-feedback setting corresponds to standard supervised learning and was thoroughly studied in the literature; which we reviewed in Chapter 2. In Chapters 3 and 4 we in fact characterized *provably-minimal* assumptions for universal learning for this full-feedback setting using the optimistic learning framework.

Optimistic learning. In this framework, the minimal assumptions are precisely those that allow for the existence of a universally consistent learning rule. First, this corresponds to characterizing the set of universally learnable processes SOAB (Strong Online Adversarial Bandits) below,

$$\text{SOAB} = \{\text{processes } \mathbb{X} : \exists \text{ learning rule } f. \text{ such that } \forall \text{ rewards within a given model, } f. \text{ is consistent}\}.$$

Second, we search for algorithms that learn under these minimal assumptions, i.e., that are universally consistent under all processes where this is possible ($\mathbb{X} \in \text{SOAB}$). These are called *optimistically universal* procedures. These optimistically universal rules, if they exist, are as general as one could hope for: they enjoy the strong property that for any given process $\mathbb{X}$, if they fail to be universally consistent, then no other algorithm would be either. We refer to Chapter 2 for more in-depth motivation and overview of the optimistic framework.

Universal learning in contextual bandits. While the literature on universal learning in the case of full feedbacks is very extensive, it is surprisingly sparse for partial feedbacks.

Previous works mostly investigated stochastic contextual bandits under important structural assumptions on rewards, such as smoothness or margin conditions. Closest to universal learning, in which one relaxes assumptions on the reward mechanism, [YZ02] showed that for continuous rewards in the contexts, strong consistency can be achieved with traditional non-parametric methods, for Euclidean context spaces. In Chapter 5, we gave the first results for contextual bandits on universal consistency per se. We focused on stationary rewards—the underlying reward mechanism is invariant over time—and showed in particular that for the main case of interest—finite action spaces $\mathcal{A}$—universal consistency is achievable under the same class of processes as for the noiseless full-feedback case (Condition SMV). Contrary to previous literature, the proposed learning rules are consistent without any assumptions on the rewards, on general spaces, and under large classes of non-i.i.d. contexts. Further, we showed that optimistically universal learning rules always exist for stationary bandits.

The present work challenges the stationarity assumption from Chapter 5. In particular, this does not allow for changes in the underlying reward mechanism, a behavior ubiquitous in current applications. It is well-known that the distribution of contexts and rewards can shift over time, such as seasonal changes in consumer behavior, and can be adversarial. Our analysis mainly focuses on two models for the strength of the adversary: oblivious rewards for which the reward mechanism can depend on the past context history, but not the past actions of the learner; and the strongest online rewards for which the rewards can be adaptive on past contexts and selected actions. This study shows that having adversarial rewards—as opposed to stationary rewards—plays a crucial role in the fundamental limits of learnability for contextual bandits, and represents a significant advancement in the general analysis of more intricate decision-making processes, such as reinforcement learning.

Related literature on contextual bandits and non-stationarity. The concept of contextual bandits was first introduced in a limited context for single-armed bandits [Woo79; Sar91]. Since then, considerable effort has been made to generalize the framework and provide efficient methods under important structural assumptions on the rewards. Most of the literature considered parametric assumptions [WKP05; LZ07; GZ09; BC+12; AC16; RS16], but substantial progress has also been achieved in the non-parametric setting towards obtaining minimax guarantees under smoothness (e.g., Lipschitz) conditions or margin assumptions [LPP09; RZ10; Sli11; PR13], with further refinements including [GJ18; RMB18].

While the above-cited works mostly focus on i.i.d. data, the non-stationary case has also been studied in the literature. The fact that the reward distribution can change over time has been widely acknowledged in the established parametric setting for contextual bandits, and has been explored under various models including [BGZ14; HMB15; KA16; Luo+18; LLS18; WIW18; Che+19]. The non-parametric case, more relevant to our work has also been considered for Lipschitz rewards and margin conditions [Sli11; SK21]. We note, however, that these works often consider non-static regret, where the baseline is also non-stationary, while we focus on the excess regret compared to *fixed* policies.

6.1.1 Summary of the present work

We mainly focus on bounded rewards. Our first main result shows that in the main case of interest of finite action spaces $\mathcal{A}$ and separable metrizable spaces $\mathcal{X}$ admitting a non-atomic

probability measure, optimistic universal learning is impossible, even under the weakest adversarial model which we call *memoryless*: rewards conditionally on their selected action and context are independent but may follow different conditional distributions. This implies that adapting algorithms for specific context processes is necessary to ensure universal learning. This is the first example of such a phenomenon for online learning, for which previously considered settings always admitted optimistically universal learning rules, including realizable (noiseless) supervised learning (Chapter 3), arbitrarily noisy or adversarial rewards in supervised learning (Chapter 4), and stationary contextual bandits (Chapter 5). Intuitively, *personalization* and *generalization* are incompatible for contextual bandits with non-stationary rewards.

Next, we study universally learnable processes for various adversarial reward models. On the negative side, we show that in the main case of interest, the set of learnable processes for stationary contextual bandits or supervised learning given by Condition SMV, is no longer fully learnable even for memoryless rewards: learning with adversarial rewards is fundamentally more difficult. This comes as a surprising result since SMV processes admitted universal learning in all previous learning settings. We further identify novel necessary and sufficient conditions, involving intricate behavior of duplicates in the context process. In particular, for memoryless, oblivious, and online rewards, the set of learnable processes is strictly between SMV and the smaller class given by Condition CS. For this same case of interest, we give an exact characterization of these learnable processes for online rewards: this characterization involves a sort of convergence rate of the instance process towards its limit distribution. Given the knowledge of this rate, universal learning is achievable with a learning rule that we provide; on the other hand, without a priori knowledge on this rate, universal learning is impossible since optimistic universal learning is not achievable. While we leave the exact characterization for memoryless and oblivious rewards as an open question for finite action spaces $\mathcal{A}$ and context spaces admitting a non-atomic probability measure, our characterizations in all other cases are complete.

Last, we give extensions of the above results, when the rewards are unbounded or satisfy some regularity constraints, namely uniform continuity.

6.1.2 New classes of stochastic processes and measure-theoretic techniques for learning theory.

We identify novel classes of processes that arise in the characterization of learnable processes. In the main case of interest, we give a new condition $\mathcal{C}_4$ that is necessary for oblivious rewards, these can be dependent on the past context history, but only on the selected action at the current time t. Informally, while SMV processes only require that the process visits only a sublinear number of sets from any countable partition of the context space $\mathcal{X}$, the necessary condition $\mathcal{C}_4$ requires this sublinear behavior to be uniform spatially in $\mathcal{X}$.

On the positive side, we introduce a novel sufficient condition $\mathcal{C}_5$ under which universal learning is possible, with CS $\subsetneq \mathcal{C}_5$ in general. Intuitively, this asks that there is a specific rate at which we can add duplicates while still preserving the CS behavior. With the knowledge of the correct rate to add duplicates, we can design universally consistent algorithms. This should be related to the property observed in Proposition 5.1 from Chapter 5, that if we

were to replace all duplicates with an arbitrary value $x_0 \in \mathcal{X}$, SMV processes would belong to CS. The $\mathcal{C}_5$ property provides an intermediary condition. Further, we show that $\mathcal{C}_5$ is also necessary for online rewards, the strongest reward model that we consider, in which rewards can be dependent on the past history of contexts and selected actions. As a result, the condition $\mathcal{C}_5$ is an exact characterization of universally learnable processes for online rewards.

Last, in an attempt to bridge the gap $\mathcal{C}_5 \subsetneq \mathcal{C}_4$ remaining for oblivious rewards, we propose a new condition $\mathcal{C}_6$ on processes, that is necessary for universal learning. At a high level, this condition constrains non-asymptotic large deviations from the empirical distribution of contexts. In the general case of context spaces $\mathcal{X}$ admitting non-atomic probability distributions, we have $\mathcal{C}_5 \subset \mathcal{C}_6 \subsetneq \mathcal{C}_4$. This shows that further uniform continuity than the $\mathcal{C}_4$ condition is necessary.

6.1.3 Outline of the chapter

In Section 6.2, we present the main definitions and some useful contextual bandit algorithms. Our main results as well as the novel classes of stochastic processes are presented in Section 6.3. We characterize when there exist optimistically universal learning rules or not in Section 6.4 and give necessary and sufficient conditions for universal learning in Section 6.5. Model extensions including continuity assumptions on the rewards are covered in Section 6.6.

6.2 Preliminaries

Let $(\mathcal{X}, \mathcal{B})$ be a separable metrizable Borel context space and $\mathcal{A}$ a separable metrizable Borel action space $\mathcal{A}$. When considering continuity assumptions, we suppose that $\mathcal{A}$ is given with a metric d. For countable action spaces, we use the discrete topology. We are interested in the following sequential contextual bandit framework: at step $t \geq 1$, the learner observes a context $X_t \in \mathcal{X}$, then selects an action $\hat{a}_t \in \mathcal{A}$ and last, receives a reward $r_t \in \mathcal{R}$ which may be stochastic. Unless mentioned otherwise, we suppose that the rewards are bounded $\mathcal{R} = [0, \bar{r}]$ and that the upper bound $\bar{r}$ is known. Hence, without loss of generality, we may pose $\bar{r} = 1$. The learner is *online* and as such, can only use the current history to select the action $\hat{a}_t$.

Definition 6.1 (Learning rule)**.** *A* learning rule *is a sequence* $f_\cdot = (f_t)_{t\geq 1}$ *of possibly randomized measurable functions* $f_t : \mathcal{X}^{t-1} \times \mathcal{R}^{t-1} \times \mathcal{X} \to \mathcal{A}$*. The action selected at* t *is* $\hat{a}_t = f_t((X_s)_{s\leq t-1}, (r_s)_{s\leq t-1}, X_t)$.

We now precise the data generation process. We suppose that the contexts $\mathbb{X} = (X_t)_{t\geq 1}$ are generated from a general stochastic process. To define the rewards, $(r_t)_{t\geq 1}$, many models for the underlying reward mechanism are possible. In Chapter 5, we considered the case of *stationary* rewards when the rewards follow a conditional distribution $P_{r|a,x}$ conditionally on the selected action $\hat{a}_t$ and the context X_t at the current time $t \geq 1$. We consider the considerably more general case of adversarial rewards. Of particular interest to the discussion of this chapter will be 1. *oblivious* rewards which correspond to the case when the learner plays a game against an adversary oblivious to the player's actions and 2. *online* rewards

when the adversary can choose rewards depending on the complete history of contexts, selected actions and received rewards. For a stochastic process $\mathbb{X}$, we will use the notation $\mathbb{X}_{\leq t} = (X_{t'})_{t' \leq t}$. Also, for a measurable set $A \in \mathcal{B}$, we will use the shorthand $\mathbb{X} \cap A = \{X_t : X_t \in A, t \geq 1\}$.

Definition 6.2 (Reward models)**.** *The reward mechanism is said to be*

- stationary (stat.) *if there is a conditional distribution $P_{r|a,x}$ such that the rewards $(r_t)_{t\geq 1}$ given their selected action a_t and context X_t are independent and follow $P_{r|a,x}$*
- memoryless *if there are conditional distributions $(P_{r|a,x,t})_{t\geq 1}$ such that $(r_t)_{t\geq 1}$ given their selected action a_t and context X_t are independent for $t \geq 1$ and respectively follow $P_{r|a,x,t}$*
- oblivious *if there are conditional distributions $(P_{r|a,\boldsymbol{x}_{\leq t}})_{t\geq 1}$ such that r_t given the selected action a_t and the past contexts $\mathbb{X}_{\leq t}$, follows $P_{r|a,\boldsymbol{x}_{\leq t}}$*
- online *if there are conditional distributions $(P_{r|\boldsymbol{a}_{\leq t},\boldsymbol{x}_{\leq t},\boldsymbol{r}_{\leq t-1}})_{t\geq 1}$ such that r_t given the sequence of selected actions $\boldsymbol{a}_{\leq t}$ and the sequence of contexts $\mathbb{X}_{\leq t}$ and received rewards $\boldsymbol{r}_{\leq t-1}$, follows $P_{r|\boldsymbol{a}_{\leq t},\boldsymbol{x}_{\leq t},\boldsymbol{r}_{\leq t-1}}$.*

We refer to all the models except for the stationary one as *adversarial*. To emphasize the dependence of the reward in the selected action and the conditional distributions, we may write $r_t(a \mid X_t)$, $r_t(a \mid \mathbb{X}_{\leq t})$, $r_t(a \mid \mathbb{X})$, and $r_t(a \mid \boldsymbol{a}_{\leq t-1}, \mathbb{X}_{\leq t}, \boldsymbol{r}_{\leq t})$ for the corresponding reward models. When the conditioning is clear from context, we may simply write $r_t(a)$ for the reward if action a is selected. The general goal in contextual bandits is to discover or approximate an optimal policy $\pi^* : \mathcal{X} \to \mathcal{A}$ if it exists. For adversarial rewards, there may not exist a single optimal policy π^*. Instead, we aim for consistent algorithms that have sublinear regret compared to any fixed measurable policy.

Definition 6.3 (Consistency and universal consistency)**.** *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$, $(r_t)_{t\geq 1}$ be a reward mechanism and f. be a learning rule. Denote by $(\hat{a}_t)_{t\geq 1}$ its selected actions. We say that f. is* consistent *under $\mathbb{X}$ with rewards r if for any measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \leq 0, \quad (a.s.).$$

We say that f. is universally consistent *for a given reward model if it is consistent under $\mathbb{X}$ with any reward within the considered reward model.*

Even in the simplest case of full-feedback noiseless learning [Han21a], universal consistency is not always achievable. For instance, if the process $\mathbb{X}$ visits a distinct instance at each step the learner, the information gathered on previous instances $\mathbb{X}_{\leq t-1}$ does not provide information on the rewards for instance X_t. We are then interested in understanding the set of processes $\mathbb{X}$ on $\mathcal{X}$ for which universal learning is possible. More practically, we aim to provide optimistically universally consistent learning rules which, if they exist, would be universally consistent whenever this is possible.

Definition 6.4 (Optimistically universal learning rule)**.** *For a given reward model that we write model* $\in \{stat, memoryless, oblivious, prescient, online\}$*, we define*

$$\textsc{Soab}_{model} = \{\mathbb{X} : \exists \text{ learning rule universally consistent for model rewards under } \mathbb{X}\}.$$

We say that a learning rule f*. is* optimistically universal *for the reward model if it is universally consistent under any process* $\mathbb{X} \in \textsc{Soab}_{model}$ *for that reward model.*

In general $\textsc{Soab}_{online} \subset \textsc{Soab}_{oblivious} \subset \textsc{Soab}_{memoryless} \subset \textsc{Soab}_{stat}$.

6.2.1 Two main classes of stochastic processes

We briefly recall the definitions of Condition CS and Condition SMV on stochastic processes arising in our characterizations of learnable processes, which we introduced in Chapter 2. Given a stochastic process $\mathbb{X}$ on $\mathcal{X}$, we first define the limit submeasure $\hat{\mu}_{\mathbb{X}}$ as follows. For any $A \in \mathcal{B}$, $\hat{\mu}_{\mathbb{X}}(A) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_A(X_t)$. Intuitively, this quantifies the (limsup) proportion of times that the process visits a set $A \in \mathcal{B}$. The first condition of stochastic processes that we introduce essentially asks that the expected empirical limsup frequency is a continuous sub-measure on $\mathcal{B}$.

Condition CS. *For every decreasing sequence* $\{A_k\}_{k=1}^{\infty}$ *of measurable sets in* $\mathcal{X}$ *with* $A_k \downarrow \emptyset$*,* $\mathbb{E}[\hat{\mu}_{\mathbb{X}}(A_k)] \underset{k\to\infty}{\longrightarrow} 0$.

This condition is useful for learning because, intuitively, it shows that one can empirically disregard the behavior of a policy on "small" sets. This then allows for an approach akin to empirical risk minimization to achieve universal consistency. We refer to Section 5.3.2 for a detailed exposition of the algorithmic details. For our purposes, and as introduced in Chapter 5, we will need to use the previous condition on *sparsified* stochastic processes which may take their defined values on a subset of possibly random times $\mathcal{T} \subseteq \mathbb{N}$ instead of the complete set of times $\mathbb{N}$, and fill the remaining times with any fixed "dummy" value $x_\varnothing \notin \mathcal{X}$. Precisely, given a process $\mathbb{X} = (X_t)_{t\geq 1}$ and an $\mathbb{X}$-dependent random set $\mathcal{T} \subseteq \mathbb{N}$, we define the sparsified process $\mathbb{X}^{\mathcal{T}}$ on $\mathcal{X} \cup \{x_\varnothing\}$ via $X_t^{\mathcal{T}} = X_t$ if $t \in \mathcal{T}$ and $X_t = x_\varnothing$ otherwise. The purpose of the non-value $x_\varnothing$ is that times $t \in \mathcal{T}$ do not contribute to empirical frequences in $\hat{\mu}_{\mathbb{X}^{\mathcal{T}}}(A)$ for sets $A \subseteq \mathcal{X}$. This leads to an *extended* definition of CS for sparsified processes with the same definition as in Condition CS: that is, $\mathbb{X}^{\mathcal{T}} \in$ CS if for every monotone sequence $A_k \downarrow \emptyset$ of measurable sets $A_k \in \mathcal{B}$, we have $\lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{\mathbb{X}^{\mathcal{T}}}(A_k)] = 0$.

The next condition asks that $\mathbb{X}$ visits a sublinear number of sets of any measurable partition of $\mathcal{X}$.

Condition SMV. *For every disjoint sequence* $\{A_k\}_{k=1}^{\infty}$ *of measurable sets of* $\mathcal{X}$ *such that* $\bigcup_{k=1}^{\infty} A_k = \mathcal{X}$*, (every countable measurable partition),* $|\{k \geq 1 : A_k \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$*,* $(a.s.)$.

Intuitively, this condition asks that the process does not keep exploring completely different regions of the space $\mathcal{X}$. It is known that even in the noiseless full-feedback setting, SMV is a necessary condition for universal learning [Han21a] since intuitively, the past history does not provide any information on newly visited regions for a learner. In fact, this is also

a sufficient condition for the noiseless full-feedback setting as we showed in Chapter 3. Both classes CS $\subset$ SMV defined above are very general classes of processes; they both include in particular i.i.d. or stationary ergodic processes.

6.2.2 Useful algorithms

We will use as subroutines in particular the same two algorithmic ingredients as in Chapter 5, which we briefly recall here. We refer to Section 5.3.2 for a detailed description and discussion of these algorithms. First, we will use the algorithm EXP3.IX proposed by [Neu15] for regret bounds with high probability in adversarial bandits, which builds upon the classic EXP3 algorithm of [Aue+02]. We restate the performance of the algorithm has the following guarantee for an adequate choice of parameters.

Theorem 5.6 (High-probability regret of EXP3.IX [Neu15])**.** *For adversarial bandits with K arms,* EXP3.IX *satisfies that, for any $\delta \in (0,1)$ and $T \geq 1$, with probability at least $1-\delta$,*

$$\max_{i\in[K]} \sum_{t=1}^{T} (r_t(a_i) - r_t(\hat{a}_t)) \leq 4\sqrt{KT \ln K} + \left(2\sqrt{\frac{KT}{\ln K}} + 1\right) \ln \frac{2}{\delta}.$$

We will always use a very simplified version of this result: there exists a universal constant $c > 0$ such that

$$\max_{i\in[K]} \sum_{t=1}^{T} (r_t(a_i) - r_t(\hat{a}_t)) \leq c\sqrt{KT \ln K} \ln \frac{1}{\delta},$$

with probability $1-\delta$ for $\delta \leq \frac{1}{2}$.

Second, we use the EXPINF algorithm from Chapter 5 which uses EXP3.IX as a subroutine to achieve sublinear regret compared to an infinite countable sequence of experts. We restate the corresponding regret bounds.

Corollary 5.1. *There is an online learning rule* EXPINF *using bandit feedback such that for any countably infinite set of experts $\{E_1, E_2, \ldots\}$ (possibly randomized), for any $T \geq 1$ and $0 < \delta \leq \frac{1}{2}$, with probability at least $1-\delta$,*

$$\max_{1\leq i\leq T^{1/8}} \sum_{t=1}^{T} (r_t(E_{i,t}) - r_t(\hat{a}_t)) \leq cT^{3/4}\sqrt{\ln T} \ln \frac{T}{\delta}.$$

where $c > 0$ is a universal constant. Further, with probability one on the learning and the experts, there exists $\hat{T}$ such that for any $T \geq 1$,

$$\max_{1\leq i\leq T^{1/8}} \sum_{t=1}^{T} (r_t(E_{i,t}) - r_t(\hat{a}_t)) \leq \hat{T} + cT^{3/4}(\ln T)^{3/2}.$$

The latter result is particularly useful to achieve universal consistency under CS processes for countable action spaces. It is known [Han21a, Lemma 24] that there exists a sequence $(\pi^l)_{l\geq 1}$ of policies $\pi : \mathcal{X} \to \mathcal{A}$ that are "empirically dense" within all measurable policies $\mathcal{X} \to \mathcal{A}$, for CS processes (see Lemma 6.4 for a formal statement). Then, using this sequence of policies as the expert set for EXPINF yields a universally consistent learning rule under CS processes. This strategy is akin to more traditional empirical risk-minimization approaches in that one aims to fit the best policy within a pre-selected set of policies.

6.3 Statement of Results

Our first main result is that for contextual bandits with adversarial rewards, for generic metric spaces $\mathcal{X}$—that admit a non-atomic probability measure, e.g., any uncountable Polish space—there never exists an optimistically universal learning rule. On the other hand, if $\mathcal{X}$ does not admit a non-atomic probability measure, optimistic learning is possible.

Theorem 6.1. *Let $\mathcal{X}$ be a separable metrizable Borel space.*

1. *Let $\mathcal{A}$ be a finite action space with $|\mathcal{A}| \geq 2$.*
   - *If $\mathcal{X}$ admits a non-atomic probability measure, there does not exist an optimistically universal learning rule for any adversarial reward model considered in Definition 6.2 (i.e., all except stationary).*
   - *Otherwise, there exists an optimistically universal learning rule for all reward models from Definition 6.2 and $\textsc{Soab}_{online} = \textsc{Soab}_{stat} = SMV$.*
2. *Let $\mathcal{A}$ be a countably infinite action space, there exists an optimistically universal learning rule for all reward models from Definition 6.2 and $\textsc{Soab}_{online} = \textsc{Soab}_{stat} = CS$.*
3. *Let $\mathcal{A}$ be an uncountable separable metrizable Borel space, then universal learning is never achievable and $\textsc{Soab}_{online} = \textsc{Soab}_{stat} = \emptyset$.*

The question of whether optimistic learning is possible for finite action spaces is answered in Section 6.4. The case of infinite action spaces is treated in Section 6.6.1. Thus, Theorem 6.1 is a concatenation of Theorems 6.4 and 6.5 and Section 6.6.1.

The fact that optimistic learning is impossible in the main case of finite action space and spaces $\mathcal{X}$ admitting a non-atomic probability measure comes in stark contrast with all learning frameworks that have been studied in the universal learning literature. Namely, for the noiseless full-feedback (Chapter 3), noisy/adversarial full-feedback (Chapter 4) and stationary partial-feedback (Chapter 5) learning frameworks, analysis showed that there always existed an optimistically universal learning rule. Precisely, the optimistically universal learning rule for stationary contextual bandits in finite action spaces provided in Chapter 5 combined two strategies:

- A strategy 0, which treats each distinct context completely separately by assigning a distinct bandit subroutine to each new instance. Informally, this corresponds to learning the optimal action for each new context without gathering population information.
- A strategy 1, in which the learning rule views context in an aggregate fashion: it tries to fit the policy that performed best on the complete historical data using learning-with-experts subroutines, from a set of pre-defined policies, akin to empirical risk-minimization approaches.

The procedure to combine these strategies estimates their performance, to implement the best strategy during pre-defined periods. We show that for adversarial rewards, balancing these two strategies is impossible. In particular, an adversarial reward mechanism can fool the estimation procedure by changing behavior between the estimation period and the implementation period. We provide some intuition of the nature of this impossibility result below.

Proof sketch for the non-existence of optimistically universal learning rules when $\mathcal{A}$ is finite. The proof involves several major steps. First, one needs to show that universal learning is achievable for a large class of processes. In particular, we show that deterministic SMV processes are learnable, where SMV is the characterization of learnable processes for supervised learning or stationary contextual bandits. This is achieved by assigning each distinct instance a multi-armed bandit learner designed to learn the best action for this instance, which corresponds to pure *personalization* (strategy 0). Next, we argue that CS processes—the characterization of learnable processes for countable action spaces $\mathcal{A}$ in stationary contextual bandits—can be learned with the same structural risk minimization approach introduced in Chapter 5 for stationary contextual bandits, which corresponds to *generalization* (strategy 1).

The main challenge is to show that one cannot universally learn both classes of processes (deterministic SMV and CS) with a unique algorithm. At the high level, we show that by contradiction, personalization and generalization are incompatible. We consider a CS-like algorithm, where instances are i.i.d. during a phase, then the same sequence is repeated many times. The reward is identical for each duplicate and has the following behavior: one *safe* action a_2 always has a relatively high reward (3/4), and an *uncertain* action a_1 has a random reward (Bernoulli $\mathcal{B}(1/2)$). We then show that because of the CS property, the algorithm needs to follow the safe action to be consistent: if it explores the uncertain action too often, the incurred loss is significant. More precisely, we show that the exploration rate of the unsafe action a_2 decays to 0. Once the algorithm reaches a certain threshold, we stop the stochastic process and consider a realization of the uncertain rewards and CS-like process. Once these are taken as deterministic, the optimal policy would be to use the action a_2 when it has a high reward, which the algorithm did not perform. Repeating this process inductively with a decaying threshold, we can show that on a deterministic SMV process, the algorithm is not universally consistent.

This negative result also provides another proof that model selection is impossible for contextual bandits. A formulation of this question was posed as a COLT 2020 open problem [FKL20]. The impossibility of model selection was then recently proved first with a switching bandit problem [MZ21]. Our results show this general impossibility in a completely different context. More precisely, Proposition 6.1 below shows that universal consistency up to a fixed error tolerance $\epsilon > 0$ is always achievable under SMV processes (which were necessary for universal learning even in the stationary case in Chapter 5). However, Theorem 6.1 implies that combining these learning rules for decaying ϵ to achieve vanishing excess error is not possible in general. The proof is given in Section 6.5.3.

Proposition 6.1. *Let $\mathcal{X}$ be a separable metrizable Borel space and $\mathcal{A}$ a finite action space. For any $\epsilon > 0$, there exists a learning rule $f_\cdot^\epsilon$ such that for any process $\mathbb{X} \in$ SMV and adversarial reward mechanism $(r_t)_{t\geq 1}$, for any measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t(\epsilon)) \leq \epsilon, \quad (a.s.),$$

where $\hat{a}_t(\epsilon)$ denotes the action selected by the learning rule at time t.

Theorem 6.1 provides the characterizations of universally learnable processes in all cases except the main case of interest when $\mathcal{A}$ is finite and $\mathcal{X}$ admits a non-atomic probability measure. Giving exact characterizations for this case is rather complex and in the following, we only give necessary conditions and sufficient conditions. These require the introduction of novel classes of stochastic processes for online learning.

6.3.1 Additional classes of stochastic processes

We first recall a significantly stronger assumption from Condition FS asking that the process only visits a finite number of distinct points. This very restrictive condition will only arise for unbounded rewards $\mathcal{R} = [0, \infty)$.

Condition FS. *The process $\mathbb{X}$ satisfies $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X} \neq \emptyset\}| < \infty \quad (a.s.)$.*

We then introduce two novel conditions on stochastic processes. Before doing so, we need to introduce some exponential time scales. Intuitively, for $\alpha > 0$, the exponential time scale at rate α is the sequence of times given by $T^k(\alpha) \approx \lfloor (1+\alpha)^k \rfloor$ for $k \geq 0$. For convenience, we will instead consider for all integers $i \geq 0$ the sequence of times $T_i^k = \lfloor 2^u(1 + v2^{-i}) \rfloor$ where $k = u2^i + v$ and $u \geq 0, 0 \leq v < 2^i$ are integers. In particular, $u = \lfloor k2^{-i} \rfloor$ and $v = k \bmod 2^i$. These times have an exponential behavior with rate oscillating between 2^{-i-1} and 2^{-i} but conveniently, they form periods $[T_i^k, T_i^{k+1})$ which become finer as i increases. For $t \geq 1$, we then define $k_i(t)$ as the index k such that $t \in [T_i^k, T_i^{k+1})$. This allows us to consider the set of times t such that X_t is the first appearance of the instance on its period,

$$\mathcal{T}^i = \{t \geq 1 : \forall T_i^{k_i(t)} \leq t' < t, X_{t'} \neq X_t\}.$$

By construction, note that $\mathcal{T}^i \subset \mathcal{T}^{i+1}$ for all $i \geq 0$. We are now ready to define the next condition which intuitively asks that the corresponding sparsified process has a CS behavior uniformly in all exponential scales.

Condition 4. *For any sequence of disjoint measurable sets $(A_i)_{i \geq 1}$ of $\mathcal{X}$, we have*

$$\lim_{i \to \infty} \mathbb{E}\left[\limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^i} \mathbb{1}_{A_i}(X_t)\right] = 0.$$

Denote by $\mathcal{C}_4$ the set of all processes $\mathbb{X}$ satisfying this condition.

Then, we define the next condition which asks that there exists a rate to include decreasing exponential scales while conserving the CS property.

Condition 5. *There exists an increasing sequence of integers $(T_i)_{i \geq 0}$ such that letting*

$$\mathcal{T} = \bigcup_{i \geq 0} \mathcal{T}^i \cap \{t \geq T_i\},$$

we have $\mathbb{X}^{\mathcal{T}} \in CS$. Denote by $\mathcal{C}_5$ the set of all processes $\mathbb{X}$ satisfying this condition.

We now introduce two new conditions on stochastic processes which we will show are necessary for some of the considered reward models. These build upon the definition of $\mathcal{C}_4$ processes. Before introducing them, we need to analyze large deviations of the empirical measure in extended CS processes. The next lemma intuitively shows that for an extended process $\mathbb{X}^{\mathcal{T}} \in \text{CS}$, for large enough time steps, one can bound the deviations of the empirical measure of a set $A \in \mathcal{B}$ compared to the limit sub-measure $\hat{\mu}_{\mathbb{X}}(A)$ uniformly in the set A.

Lemma 6.1. *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$ and $\mathcal{T}$ some $\mathbb{X}$-dependent random times such that $\mathbb{X}^{\mathcal{T}} \in CS$. Then, for any $\epsilon > 0$, there exists $T_\epsilon \geq 1$ and $\delta > 0$ such that for any measurable set $A \in \mathcal{B}$,*

$$\mathbb{E}[\hat{\mu}_{\mathbb{X}^{\mathcal{T}}}(A)] \leq \delta \Longrightarrow \mathbb{E}\left[\sup_{T \geq T_\epsilon} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}} \mathbb{1}_A(X_t)\right] \leq \epsilon.$$

The proof is given in Section 6.5.1. Now consider a process $\mathbb{X} \in \mathcal{C}_4$. For any integer $p \geq 0$, the definition of $\mathcal{C}_4$ implies $\mathbb{X}^p := \mathbb{X}^{\mathcal{T}^p} \in \text{CS}$. Indeed, the sets $\mathcal{T}^i$ are increasing in $i \geq 0$, hence for $i \geq p$ one has $\mathcal{T}^p \subset \mathcal{T}^i$. As a result, the condition $\mathcal{C}_4$ implies that for any disjoint measurable sets $(A_i)_{i \geq 1}$, one has $\mathbb{E}[\hat{\mu}_{\mathbb{X}^p}(A_i)] = \mathbb{E}[\limsup_{T \to \infty} \sum_{t \leq T, t \in \mathcal{T}^p} \mathbb{1}_{A_i}(X_t)] \to 0$ as $i \to \infty$. Now for any $\epsilon > 0$ and $T \geq 1$, we define

$$\delta^p(\epsilon; T) := \sup \Bigg\{ 0 \leq \delta \leq 1 : \forall A \in \mathcal{B} \text{ s.t. } \sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A)] \leq \delta, \\ \forall \tau \geq T \text{ online stopping time}, \quad \mathbb{E}\left[\frac{1}{2\tau} \sum_{\tau \leq t < 2\tau, t \in \mathcal{T}^p} \mathbb{1}_A(X_t)\right] \leq \epsilon \Bigg\},$$

where the τ is a stopping time with respect to the filtration generated by the instance process $\mathbb{X}$. In particular, τ can be seen as an online procedure that decides when to count the number of instances of $\mathbb{X}^p$ falling in the considered set A. Note that $\delta^p(\epsilon; T)$ satisfies the property that for all measurable set A satisfying $\sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A)] \leq \delta^p(\epsilon; T)$ and any stopping time $\tau \geq T$,

$$\mathbb{E}\left[\frac{1}{2\tau} \sum_{\tau \leq t < 2\tau, t \in \mathcal{T}^p} \mathbb{1}_A(X_t)\right] \leq \epsilon,$$

which can be checked for all sets $A \in \mathcal{B}$ separately. Next, the quantity $\delta^p(\epsilon; T)$ is non-decreasing in T. Further, as a direct application of Lemma 6.1, because $\mathbb{X}^p \in \text{CS}$, there exists $T^p(\epsilon) \geq 1$ and $\delta > 0$ such that for $T \geq T^p(\epsilon)$, we have $\delta^p(\epsilon; T) \geq \delta$. As a result, we have $\delta^p(\epsilon) := \lim_{T \to \infty} \delta^p(\epsilon; T) \geq \delta > 0$. Also, the quantity $\delta^p(\epsilon; T)$ is non-increasing in p since the sets $\mathcal{T}^p$ are non-decreasing with p. Thus, $\delta^p(\epsilon)$ is also non-increasing in p. We are now ready to introduce the condition on stochastic processes based on the limit of the quantities $\delta^p(\epsilon)$.

Condition 6. *$\mathbb{X} \in \mathcal{C}_4$ and for any $\epsilon > 0$, we have $\lim_{p \to \infty} \delta^p(\epsilon) > 0$. Denote by $\mathcal{C}_6$ the set of all processes $\mathbb{X}$ satisfying this condition.*

Intuitively, this asks that the maximum deviations are also bounded in p, hence $\mathcal{C}_6$ processes have more regularity than general $\mathcal{C}_4$ processes. However, the maximum deviations are limited by the fact that they should be discernible through an online stopping time τ.

Learning setting	Stationary contextual bandits [Chapter 5]		Contextual bandits with adversarial rewards [Chapter 6]	
	$\textsc{Soab}_{stat}$	OL?	Necessary and sufficient conditions on $\textsc{Soab}$	OL?
Finite $\mathcal{A}$, $\lvert\mathcal{A}\rvert \geq 2$, $\mathcal{X}$ with non-atomic proba. measure	SMV	Yes	$\text{CS} \subsetneq \mathcal{C}_5 \subset \textsc{Soab} \subsetneq \text{SMV}$ $\mathcal{C}_5 = \textsc{Soab}_{online} \subset \textsc{Soab}_{oblivious} \subset \mathcal{C}_6$	No
Finite $\mathcal{A}$, $\lvert\mathcal{A}\rvert \geq 2$, $\mathcal{X}$ without non-atomic proba. measure	SMV	Yes	$\textsc{Soab} = \text{SMV}$	Yes
Countably infinite $\mathcal{A}$	CS	Yes	$\textsc{Soab} = \text{CS}$	Yes
Uncountable $\mathcal{A}$	$\emptyset$	N/A	$\textsc{Soab} = \emptyset$	N/A

Table 6.1: Characterization of learnable processes for universal learning in contextual bandits, depending on the action space $\mathcal{A}$, context space $\mathcal{X}$, and reward model. When the model is not specified, $\textsc{Soab}$ refers to any of the considered models. OL? = Is optimistic learning possible?

The following inclusions hold FS $\subset$ CS $\subset \mathcal{C}_5 \subset \mathcal{C}_6 \subset \mathcal{C}_4 \subset$ SMV. Indeed, the inclusion FS $\subset$ CS is known [Han21a]. CS $\subset \mathcal{C}_5$ and $\mathcal{C}_6 \subset \mathcal{C}_4$ are immediate from the definition of Condition 5 and Condition 6 respectively. The inclusion $\mathcal{C}_4 \subset$ SMV is shown in Proposition 6.6. Last, the fact that for oblivious rewards, $\mathcal{C}_6$ is necessary (Theorem 6.8) and $\mathcal{C}_5$ is sufficient (Theorem 6.12) shows that $\mathcal{C}_5 \subset \mathcal{C}_6$.

6.3.2 Necessary and sufficient conditions for universal learning

Our second main contribution is giving necessary and sufficient conditions for universal learning with adversarial rewards, which are summarized in Section 6.3.1. In addition to characterizations from Theorem 6.1, we have the following.

Theorem 6.2. *Let $\mathcal{X}$ be a separable metrizable Borel space admitting a non-atomic probability measure and $\mathcal{A}$ a finite action space with $\lvert\mathcal{A}\rvert \geq 2$. Then CS $\subsetneq \mathcal{C}_5 = \textsc{Soab}_{online} \subset \textsc{Soab}_{oblivious} \subset \textsc{Soab}_{memoryless} \subsetneq$ SMV. Further, $\textsc{Soab}_{oblivious} \subset \mathcal{C}_6 \subsetneq$ SMV.*

These results are proved in Section 6.5. The fact that $\textsc{Soab}_{memoryless} \subsetneq$ SMV is proved in Theorem 6.7. $\textsc{Soab}_{oblivious} \subset \mathcal{C}_6$ is proved in Theorem 6.8 while $\mathcal{C}_6 \subsetneq$ SMV comes from Theorem 6.7 and the fact that $\mathcal{C}_6 \subset \mathcal{C}_4$ (Theorem 6.9 further gives an example of processes in $\mathcal{C}_4 \setminus \mathcal{C}_6$). $\textsc{Soab}_{online} \subset \mathcal{C}_5$ is proved in Theorem 6.11 and CS $\subsetneq \mathcal{C}_5 \subset \textsc{Soab}_{online}$ is proved in Theorem 6.12 and Proposition 6.7. Here is the overview of relations we show between the classes of processes: for $\mathcal{X}$ admitting non-atomic probability measures, CS $\subsetneq \mathcal{C}_5 \subset \mathcal{C}_6 \subsetneq \mathcal{C}_4 \subsetneq$ SMV. We provide below an overview of the techniques we use to prove this result.

Proof techniques and intuitions. We start with the necessary conditions. To give intuitions, we focus on the weaker necessary condition $\textsc{Soab}_{oblivious} \subset \mathcal{C}_4$. This also follows the order of arguments in the complete proof. Condition $\mathcal{C}_4$ requires that the process visits a sublinear number of sets from any countable partition, uniformly in all exponential time scales $\mathcal{T}^i$ for $i \geq 1$. When this is not satisfied by a process $\mathbb{X}$, one can take advantage of these discrepancies between time scales with adversarial rewards to obtain a somewhat similar personalization/generalization incompatibility phenomenon as the one described for the non-existence of optimistically universal learning rules. More precisely, if $\mathbb{X} \notin \mathcal{C}_4$, there

exist disjoint sets $(A_j)_{j\geq 1}$ with the following behavior: the set A_j is visited infinitely often with a constant fraction of times for some time scale $\mathcal{T}^{i(j)}$. One can then consider similar oblivious rewards as before: the rewards are identical on duplicates but with one safe and one uncertain option. The safe action always has a reward 3/4 while the uncertain action has a reward sampled from a Bernoulli $\mathcal{B}(1/2)$. To avoid a constant error rate, for each set A_i, eventually, the algorithm's exploration rate of the uncertain action decays to 0. Once this rate is sufficiently small, we stop the process and consider a specific realization of the rewards. Since the rewards are now deterministic, the optimal policy on A_j selects the uncertain action when beneficial, yielding a non-negligible improvement in average reward compared to the algorithm. Because the sets A_j are disjoint, we can repeat this argument for the decreasing scales of exponential times $\mathcal{T}^{i(j)}$ and concatenate the obtained policies. Compared to the concatenated policy, the algorithm incurs non-negligible regret infinitely often for the constructed rewards, and the algorithm is not universally consistent as a result. We omitted for simplicity details including the fact that during each constructed phase, no information on the rewards of future local space zones should be revealed. This can be achieved with oblivious rewards thanks to their dependence on the past history of contexts, but would not be possible with stationary rewards. This was expected since for stationary rewards, we showed in Chapter 5 that universal learning under all SMV processes is possible, hence $\mathcal{C}_4$ is not necessary.

For sufficient conditions, we show that universal learning is possible under $\mathcal{C}_5$ processes with adversarial rewards. This condition asks that there exist an adequate rate to add duplicates (within the exponential time scales $\mathcal{T}^i$), that still conserves the CS property: the set $\mathcal{T}$ of times from Condition 5. For all points in this set $\mathcal{T}$, we can use the structural risk minimization approach (strategy 1) since these points still have CS behavior. For the remaining duplicates, we use pure personalization by assigning a bandit learner to each distinct instance (strategy 0). A caveat with this approach is that strategy 0 performs well when a single instance has many duplicates, while $\mathcal{T}$ only accounts for points with one duplicate per period within the current exponential time scale. To solve this issue, our complete learning rule uses batching techniques to appropriately balance strategies 0 and 1 (morally allowing for an increasing number of duplicates per period with respect to the exponential scales given by $\mathcal{T}$). We note that it heavily relies on the knowledge of the correct rate to add duplicates for $\mathbb{X}$, as expected given that optimistically universal learning rules do not exist.

In particular, our characterization is complete for the strongest online rewards, unlike for memoryless and oblivious rewards. We believe that $\mathcal{C}_5 \subsetneq \mathcal{C}_6$ in general. The proof of Theorem 6.8 for the necessity of $\mathcal{C}_6$ for oblivious rewards can be tightened given a stronger reward model in which the reward adversary can additionally take into account the *complete* sequence $\mathbb{X}$—instead of the revealed contexts to the learner $\mathbb{X}_{\leq t}$. We refer to this reward model as *prescient* rewards (see Definition 6.5 for a formal definition) and show that in this case, a stronger $\mathcal{C}_7$ condition is necessary (Theorem 6.10). We leave open the question of whether $\mathcal{C}_5 = \mathcal{C}_7$. If this were true, then we also have an exact characterization for prescient rewards.

Our findings are summarized in Table 1, which also compares learnable processes for stationary and adversarial contextual bandits. We leave open the exact characterization

of learnable processes for memoryless and oblivious rewards in finite action spaces $\mathcal{A}$ and context spaces admitting a non-atomic probability measure.

Open question: *Let $\mathcal{X}$ be a separable metrizable Borel space admitting a non-atomic probability measure and $\mathcal{A}$ a finite action space with $|\mathcal{A}| \geq 2$. What is an exact characterization of $\text{SOAB}_{memoryless}$ or $\text{SOAB}_{oblivious}$?*

Finally, we also give results in a setting where we assume that rewards are unbounded. We answer the same questions: What are the learnable processes for which universal learning is possible, and can we obtain optimistically universal learning rules? We use a subscript $\text{SOAB}^{unbounded}$ to specify that we consider the case of unbounded rewards. We show that in that case, results are identical to the case of stationary contextual bandits.

Proposition 6.2. *Let $\mathcal{X}$ be a separable metrizable Borel space and consider unbounded rewards. For all reward models,*

- *if $\mathcal{A}$ is countable, $\text{SOAB}^{unbounded} = FS$. Further, there is an optimistically universal learning rule,*
- *if $\mathcal{A}$ is uncountable, universal learning is never achievable.*

The proof is given in Section 6.6.2. Last, we extend our results to rewards with additional regularity assumptions. For a given metric d on $\mathcal{A}$, we suppose that they are uniformly-continuous, generalizing a notion introduced in Chapter 5.

Let $(\mathcal{A}, d)$ be a separable metric space. The reward mechanism $(r_t)_{t\geq 1}$ is uniformly-continuous if for any $\epsilon > 0$, there exists $\Delta(\epsilon) > 0$ such that

$$\forall t \geq 1, \forall (\boldsymbol{x}_{\leq t}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}) \in \mathcal{X}^t \times \mathcal{A}^{t-1} \times \mathcal{R}^{t-1}, \forall a, a' \in \mathcal{A},$$
$$d(a, a') \leq \Delta(\epsilon) \Rightarrow |\mathbb{E}[r_t(a) - r_t(a') \mid \mathbb{X}_{\leq t} = \boldsymbol{x}_{\leq t}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}]| \leq \epsilon,$$

For uniformly-continuous rewards, we use a reduction to the case of rewards without regularity assumptions, which we refer to as *unrestricted* rewards. Then, we recover the same results for uniformly-continuous rewards, in totally-bounded (resp. non-totally-bounded) action spaces as for unrestricted rewards in finite (resp. countably infinite) action spaces. We adopt the notation SOAB-UC to emphasize that we consider uniformly-continuous rewards.

Theorem 6.3. *Let $\mathcal{X}$ be a metrizable Borel space and select a model within $model \in \{memoryless, oblivious, online\}$.*

- *If $\mathcal{A}$ is a totally-bounded metric space, all properties for SOAB_{model} for finite action spaces described in Theorem 6.2 hold for SOAB-UC_{model}. Further, there is an optimistically universal learning rule for uniformly-continuous rewards if and only if there is one for finite action spaces for unrestricted rewards as in Theorem 6.1.*
- *If $\mathcal{A}$ is a non-totally-bounded metric space, all properties for SOAB_{model} for countable action spaces described in Theorem 6.1 hold for SOAB-UC_{model}. Further, there is always an optimistically universal learning rule for uniformly-continuous rewards.*

This result is proved in Section 6.6.3 and is a concatenation of Proposition 6.8 for necessary conditions and Theorem 6.13 and Theorem 6.14 for sufficient conditions for universal learning.

6.4 Existence or Non-Existence of an Optimistically Universal Learning Rule

In this section, we ask the question of whether there exists an optimistically universal learning rule for finite action spaces. In fact, in all the frameworks considered for universal learning—noiseless (Chapter 3) or noisy/adversarial responses (Chapter 4) in the full-feedback setting and stationary partial-feedback responses (Chapter 5)—analysis showed that optimistically universal learning always existed. However, the learning rule provided in Chapter 5 for stationary rewards under SMV processes heavily relies on the assumption that the rewards are stationary to make good estimates of the performance of different learning strategies. In particular, one can easily check that this learning rule would not be universally consistent under adversarial rewards even in the weakest memoryless setting. Instead, we will show that for contextual bandits with adversarial rewards, in general, there do not exist optimistically universal learning rules.

To do so, we first need to argue that the set of learnable processes even in the online setting $\textsc{Soab}_{online}$ contains a reasonably large class of processes. We first show that using the EXP3.IX algorithm for adversarial bandits [Neu15] as a subroutine yields a universally consistent learning rule for processes $\mathbb{X}$ which visit a sublinear number of distinct instances.

Proposition 6.3. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathcal{A}$ a finite action space. There exists a learning rule which is universally consistent for online rewards under any process $\mathbb{X}$ satisfying $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$ $(a.s.)$.*

Proof Consider the learning rule $f.$ that simply performs independent copies of the EXP3.IX algorithm in parallel, so that each distinct instance visited is assigned a EXP3.IX. More precisely, for any $t \geq 1$, instances $\boldsymbol{x}_{\leq t}$ and observed rewards $\boldsymbol{r}_{\leq t-1}$, we define

$$f_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}, x_t) = \text{EXP3.IX}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t}),$$

where $S_t = \{t' < t : x_{t'} = x_t\}$ is the set of times that x_t was visited previously and $\hat{a}_{t'}$ denotes the action selected at time t' for $t' < t$. We now show that this learning rule is universally consistent on any process $\mathbb{X}$ which visits a sublinear number of distinct instances almost surely. For simplicity, we denote $\hat{a}_t$ the action selected by $f.$ at time t. Let $\mathbb{X}$ such that almost surely, $\frac{1}{T}|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| \to 0$. Denote by $\mathcal{E}$ this event, and for any $T \geq 1$ we define $\epsilon(T) = \frac{1}{T}|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}|$ and $S_T = \{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}$, hence $|S_T| = T\epsilon(T)$. Further, for any $x \in S_T$ we pose $\mathcal{T}_T(x) = \{t \leq T : X_t = x\}$. Let $\mathcal{H}_0(T) = \{x \in S_T : |\mathcal{T}_T(x)| < \frac{1}{\sqrt{\epsilon(T)}}\}$, $\mathcal{H}_1(T) = \{x \in S_T : \frac{1}{\sqrt{\epsilon(T)}} \leq |\mathcal{T}_T(x)| < \ln^2 T\}$ and $\mathcal{H}_2(T) = \{x \in S_T : |\mathcal{T}_T(x)| \geq \ln^2 T\}$, so that $S_T = \mathcal{H}_0(T) \cup \mathcal{H}_1(T) \cup \mathcal{H}_2(T)$. Note that

$$\sum_{x \in \mathcal{H}_0(T)} \sum_{t \in \mathcal{T}_T(x)} r_t(\pi(X_t)) - r_t(\hat{a}_t) \leq \frac{|\mathcal{H}_0(T)|}{\sqrt{\epsilon(T)}} \leq \sqrt{\epsilon(T)}T.$$

Now fix a measurable policy $\pi : \mathcal{X} \to \mathcal{A}$. Then,

$$\sum_{x \in \mathcal{H}_2(T)} \sum_{t \in \mathcal{T}_T(x)} r_t(\pi(X_t)) - r_t(\hat{a}_t) \leq \sum_{x \in \mathcal{H}_2(T)} \max_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}_T(x)} (r_t(a) - r_t(\hat{a}_t)).$$

Now recall that for any $x \in S_T$, on $\mathcal{T}_T(x)$ the algorithm EXP3.IX was performed. As a result, by Theorem 5.6, conditionally on the realization $\mathbb{X}$, for any $x \in \mathcal{H}_2(T)$, with probability $1 - \frac{1}{T^3}$, conditionally on $\mathbb{X}$,

$$\max_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}_T(x)} (r_t(a) - r_t(\hat{a}_t)) \leq 3c\sqrt{|\mathcal{A}||\mathcal{T}_T(x)| \ln |\mathcal{A}|} \ln T \leq |\mathcal{T}_T(x)| \cdot 3c \frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{\ln T}.$$

Noting that $|\mathcal{H}_2(T)| \leq T$, we obtain by the union bound that (conditionally on $\mathbb{X}$) with with probability $1 - \frac{1}{T^2}$,

$$\sum_{x \in \mathcal{H}_2(T)} \max_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}_T(x)} (r_t(a) - r_t(\hat{a}_t)) \leq 3c \frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{\ln T} \sum_{x \in \mathcal{H}_2(T)} |\mathcal{T}_T(x)| \leq 3c\sqrt{|\mathcal{A}| \ln |\mathcal{A}|} \frac{T}{\ln T}.$$

We denote by $\mathcal{F}_T$ the event when the above equation holds. We have $\mathbb{P}[\mathcal{F}_T] \geq 1 - \frac{1}{T^2}$ where the probability is also taken over $\mathbb{X}$. We now turn to points in $\mathcal{H}_1(T)$ for which we need to go back to the proof of Theorem 5.6 from [Neu15]. Taking the same notations as in the original proof, for $u \geq 1$, let $\eta_u = 2\gamma_u = \sqrt{\frac{\ln |\mathcal{A}|}{|\mathcal{A}| u}}$, and for any $t \geq 1$, $a \in \mathcal{A}$ denote by $p_{t,a}$ the probability that the learning rule selects action a at time t, and let $\ell_{t,a} = 1 - r_t(a)$. Next, let $u(t) = |\{s \leq t : X_s = X_t\}|$ and pose $\tilde{\ell}_{t,a} = \frac{1 - r_t(a)}{p_{t,a} + \gamma_u} \mathbb{1}[\hat{a}_t = a]$. Using the derivations of the proof of Theorem 5.6, for any $x \in S_T$, writing $\mathcal{T}_T(x) = \{t_1(x), \ldots, t_{|\mathcal{T}_T(x)|}\}$, for any $a' \in \mathcal{A}$,

$$\sum_{u=1}^{|\mathcal{T}_T(x)|} \left(\ell_{t_u, \hat{a}} - \tilde{\ell}_{t_u, a'} \right) \leq \frac{\ln |\mathcal{A}|}{\eta_{|\mathcal{T}_T(x)|}} + \sum_{u=1}^{|\mathcal{T}_T(x)|} \eta_u \sum_{a \in \mathcal{A}} \tilde{\ell}_{t_u, a}.$$

Summing these equations with $a' = \pi(x)$, we obtain

$$\sum_{x \in \mathcal{H}_1(T)} \sum_{t \in \mathcal{T}_T(x)} (1 - \tilde{\ell}_{t, \pi(X_t)}) - r_t(\hat{a}_t) \leq \sum_{x \in \mathcal{H}_1(T)} \sqrt{|\mathcal{A}| \ln |\mathcal{A}| |\mathcal{T}_T(x)|} + \sum_{x \in \mathcal{H}_1(T)} \sum_{t \in \mathcal{T}_T(x)} \eta_{u(t)} \sum_{a \in \mathcal{A}} \tilde{\ell}_{t,a}.$$

Now let for any $a \in \mathcal{A}$, conditionally on $\mathbb{X}$, the sequence $(\sum_{x \in \mathcal{H}_1(T')} \sum_{t \in \mathcal{T}_{T'}(x)} \eta_{u(t)} (\tilde{\ell}_{t,a} - \ell_{t,a}))_{T' \leq T}$ is a super-martingale (the immediate expected value of $\tilde{\ell}_{t,a}$ is $\frac{p_{u(t)}}{p_{u(t)} + \gamma_{u(t)}} \ell_{t,a}$) and each increment is upper-bounded by 2 in absolute value: $0 \leq \eta_{u(t)} \tilde{\ell}_{t,a} \leq \eta_{u(t)} \frac{\ell_{t,a}}{p_{u(t),a} + \gamma_{u(t)}} \leq \frac{\eta_{u(t)}}{\gamma_{u(t)}} \leq 2$. Therefore, Azuma's inequality implies

$$\mathbb{P}\left[\sum_{x \in \mathcal{H}_1(T)} \sum_{t \in \mathcal{T}_T(x)} \eta_{u(t)} \sum_{a \in \mathcal{A}} (\tilde{\ell}_{t,a} - \ell_{t,a}) \leq 4T^{3/4} \mid \mathbb{X} \right] \geq 1 - e^{-2\sqrt{T}}.$$

Similarly, because $0 \leq \tilde{\ell}_{t,a} \leq \frac{1}{\gamma_{u(t)}} = 2\sqrt{\frac{|\mathcal{A}|u(t)}{\ln|\mathcal{A}|}}$, we have

$$\mathbb{P}\left[\sum_{x\in\mathcal{H}_1(T)}\sum_{t\in\mathcal{T}_T(x)}\sum_{a\in\mathcal{A}}(\tilde{\ell}_{t,\pi(X_t)} - \ell_{t,\pi(X_t)}) \leq 4\sqrt{\frac{|\mathcal{A}|}{\ln|\mathcal{A}|}}T^{3/4}\ln T \mid \mathbb{X}\right] \geq 1 - e^{-2\sqrt{T}}.$$

As a result, on an event $\mathcal{G}_T$ of probability at least $1-(1+|\mathcal{A}|)e^{-2\sqrt{T}}$, we have

$$\begin{aligned}
\sum_{x\in\mathcal{H}_1(T)}\sum_{t\in\mathcal{T}_T(x)} r_t(\pi(X_t)) - r_t(\hat{a}_t) &\leq \sum_{x\in\mathcal{H}_1(T)}\sqrt{|\mathcal{A}|\ln|\mathcal{A}||\mathcal{T}_T(x)|} + \sum_{x\in\mathcal{H}_1(T)}\sum_{t\in\mathcal{T}_T(x)}\eta_{u(t)}\sum_{a\in\mathcal{A}}\ell_{t,a} \\
&\qquad + 4\sqrt{\frac{|\mathcal{A}|}{\ln|\mathcal{A}|}}T^{3/4}\ln T + 4T^{3/4} \\
&\leq \sum_{x\in\mathcal{H}_1(T)}\sqrt{|\mathcal{A}|\ln|\mathcal{A}||\mathcal{T}_T(x)|} + \sum_{x\in\mathcal{H}_1(T)}|\mathcal{A}|\sum_{t\in\mathcal{T}_T(x)}\eta_{u(t)} \\
&\qquad + 4\sqrt{\frac{|\mathcal{A}|}{\ln|\mathcal{A}|}}T^{3/4}\ln T + 4T^{3/4} \\
&\leq \sum_{x\in\mathcal{H}_1(T)} 3\sqrt{|\mathcal{A}|\ln|\mathcal{A}||\mathcal{T}_T(x)|} + 8\sqrt{|\mathcal{A}|}T^{3/4}\ln T \\
&\leq 3\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\epsilon(T)^{1/4}T + 8\sqrt{|\mathcal{A}|}T^{3/4}\ln T.
\end{aligned}$$

Combining all our estimates, we showed that on $\mathcal{F}_T \cap \mathcal{G}_T$,

$$\sum_{t\leq T} r_t(\pi(X_t)) - r_t(\hat{a}_t) \leq 8|\mathcal{A}|T^{3/4}\ln T + 3c\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\frac{T}{\ln T} + (\sqrt{\epsilon(T)} + 3\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\epsilon(T)^{1/4})T$$

Now note that $\sum_{T\geq 1}\mathbb{P}[\mathcal{F}_T^c] + \mathbb{P}[\mathcal{G}_T^c] < \infty$. Hence, the Borel-Cantelli lemma implies that on an event $\mathcal{A}$ of probability one, there exists $\hat{T} \geq 1$ such that for any $T \geq \hat{T}$, the event $\mathcal{F}_T \cap \mathcal{G}_T$ is satisfied. As a result, on the event $\mathcal{E} \cap \mathcal{A}$, since $\epsilon(T) \to 0$, we obtain

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - r_t(\hat{a}_t) \leq 0.$$

By union bound, $\mathcal{E} \cap \mathcal{A}$ has probability one, hence we proved that the learning rule *f.* is universally consistent on $\mathbb{X}$. This ends the proof of the proposition. ■

As a simple consequence of Proposition 6.3, deterministic SMV processes are always universally learnable even in the online rewards setting.

Proposition 6.4. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathcal{A}$ a finite action space. There exists a learning rule that is universally consistent for any deterministic process $\mathbb{X} \in$ SMV under online rewards.*

Proof We first show that any deterministic process $\mathbb{X} \in$ SMV visits a sublinear number of distinct instances almost surely. Denote $S_T = \{X_t : t \leq T\}$ the set of visited instances until

time T and let $S = \bigcup_{T\to\infty} S_T$. Then, $\{x\}_{x\in S}$ forms a countable sequence of disjoint sets. Hence, by the SMV property and because $\mathbb{X}$ is deterministic, we have that

$$|\{x : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = |S_t| = |\{x \in S : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T), \quad (a.s.).$$

Hence, by Proposition 6.3, the learning rule which performs EXP3.IX independently for each distinct visited instance is universally consistent under $\mathbb{X}$. This ends the proof of the proposition. ■

Next, we argue that CS processes are also universally learnable in the online rewards setting. In the case of countable action sets $\mathcal{A}$, in Chapter 5 we gave a universally consistent learning rule EXPINF under CS processes using Corollary 5.1. Precisely, the learning rule uses a result from [Han21a] showing that there exists a countable set of policies $\Pi = \{\pi^i : \mathcal{X} \to \mathcal{A}, i \geq 1\}$ that is empirically dense within measurable policies under any CS process. As a result, to yield a universally consistent learning rule under CS processes, it suffices to have a learning rule with sublinear regret compared to any policy $\pi \in \Pi$. The algorithm EXPINF achieves this property using restarted EXP3.IX subroutines with a slowly increasing finite set of experts from the sequence Π. Because the subroutines EXP3.IX have guarantees in the adversarial bandit framework, EXPINF directly inherits this guarantee and is a result universally consistent under CS processes for online rewards. Thus, CS $\subset$ SOAB$_{online}$.

We are now ready to show that for spaces $\mathcal{X}$ on which there exists a non-atomic probability measure on the space $\mathcal{X}$, there does not exist any optimistically universally consistent learning rule. Precisely, we show that there is no learning rule that is universally consistent both on CS and deterministic SMV processes. Note that most context spaces $\mathcal{X}$ of interest would admit a non-atomic probability measure, in particular any uncountable Polish space.

Theorem 6.4. *Let $\mathcal{X}$ a metrizable separable Borel space such that there exists a non-atomic probability measure μ on $\mathcal{X}$, i.e., such that $\mu(\{x\}) = 0$ for all $x \in \mathcal{X}$. If $\mathcal{A}$ is a finite action space with $|\mathcal{A}| \geq 2$, then there does not exist an optimistically universal learning rule for memoryless rewards (a fortiori for oblivious, prescient, or online rewards).*

Proof We fix $a_1, a_2 \in \mathcal{A}$ two distinct actions. Suppose that there exists an optimistically universal learning rule $f.$. For simplicity, we will denote by $\hat{a}_t$ the action chosen by this learning rule at step t. We will construct a deterministic process $\mathbb{X} \in$ SMV and rewards r_t for which $f.$ does not achieve universal consistency.

We construct the process $\mathbb{X}$ and rewards $(r_t)_{t\geq 1}$ recursively. Let $\epsilon_k = 2^{-k}$ for $k \geq 1$. The process and rewards are constructed together with times T_k such that a significant regret is incurred to the learner between times T_k and T_{k+1} for all $k \geq 1$. We pose $T_0 = 0$. We are now ready to start the induction. Suppose that we have already defined T_l for $l < k$ and the deterministic process $\mathbb{X}_{\leq T_{k-1}}$ as well as the deterministic rewards r_t for $t \geq T_{k-1}$. Let $\mathbb{Z} = (Z_i)_{i\geq 1}$ be an i.i.d. sequence on $\mathcal{X}$ with distribution μ. Pose $T^i = \frac{(1+i)!}{\epsilon_k} T_{k-1}$ for $i \geq 0$ and $k_i = \epsilon_k T^i \, (= (1+i)! T_{k-1})$, $n_i = \sum_{j<i} k_j$ for $i \geq 0$. Letting $\bar{x} \in \mathcal{X}$ an arbitrary instance, we now consider the following process $\tilde{\mathbb{X}}$:

$$\tilde{X}_t = \begin{cases} X_t, & t \le T_{k-1}, \\ \bar{x}, & T_{k-1} < t < T^0, \\ Z_{n_i+l}, & t = T^i + p \cdot k_i + l, \quad 0 \le p < \frac{1}{\epsilon_k},\ 0 \le l < k_i, i \ge 0, \\ \bar{x}, & 2T^i \le t < T^{i+1}, \quad i \ge 0. \end{cases}$$

The process is deterministic until time T^0. From this point, the process is constructed by periods, where period $i \ge 0$ corresponds to times $T^i \le t < T^{i+1} = (1+i)T^i$. Each period i has a first phase $T^i \le t < 2T^i$ composed of $\frac{1}{\epsilon_k}$ sub-phases of length $k_i = \epsilon_k T^i$ on which the process repeats exactly. We can therefore focus on the first sub-phase $T^i \le t < T^i(1+\epsilon_k)$, which is constructed as an i.i.d. process following distribution μ independent from the past samples. In the second phase of period i for $2T^i \le T^{i+1}$ the process is idle equal to $\bar{x}$. This ends the construction of the process $\tilde{\mathbb{X}}$.

We now argue that $\tilde{\mathbb{X}} \in \mathrm{CS}$. Indeed, note that forgetting about the part for $t \le T^0$, and idle phases where the process visits $\bar{x}$ only, this process takes values from an i.i.d. process $\mathbb{Z}$ and each value is duplicated $\frac{1}{\epsilon_k}$ times throughout the whole process. Formally, let $(A_p)_{p\ge1}$ be a decreasing sequence of measurable sets with $A_p \downarrow \emptyset$. Then for any $T^i < T \le T^{i+1}$ with $i \ge 1$ we have, for p sufficiently large so that $\bar{x} \notin A_p$,

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{A_p}(\tilde{X}_t) &\le \frac{2T^{i-1}}{T^i} + \frac{1}{\epsilon_k T^i}\sum_{l=n_i}^{n_i+k_i-1} \mathbb{1}_{A_p}(Z_l) \\ &\le \frac{2}{1+i} + \frac{n_i+k_i}{k_i}\frac{1}{n_i+k_i}\sum_{l=0}^{n_i+k_i-1} \mathbb{1}_{A_p}(Z_l). \end{aligned}$$

Last, we note that $\frac{n_i+k_i}{k_i} \to 1$ as $i \to \infty$. As a result, we obtain $\hat{\mu}_{\tilde{\mathbb{X}}}(A_p) \le \hat{\mu}_{\mathbb{Z}}(A_p)$. Because $\mathbb{Z} \in \mathrm{CS}$, we have $\mathbb{E}[\hat{\mu}_{\mathbb{Z}}(A_p)] \to 0$ as $p \to \infty$, which proves $\mathbb{E}[\hat{\mu}_{\tilde{\mathbb{X}}}(A_p)] \to 0$ as well. This ends the proof that $\tilde{\mathbb{X}} \in \mathrm{CS}$.

We now construct rewards. Before doing so, for any $i \ge 0$, let δ_i such that

$$\mathbb{P}\left[\min_{1\le u<v<n_{i+1}} \rho(Z_i, Z_j) \le \delta_i\right] \le 2^{-i-2}.$$

This is possible because μ is non-atomic, as a result with probability one, all Z_k for $k \ge 1$ are distinct. Then, by the union bound, with probability at least $1 - \frac{1}{2} = \frac{1}{2}$, for all $i \ge 0$ we have

$$\min_{1\le u<v<n_{i+1}} \rho(Z_u, Z_v) > \delta_i.$$

We denote by $\mathcal{E}$ the event where the above inequality holds for all $i \ge 1$ and all $u \ge 1$, $Z_u \ne \bar{x}$. Because μ is non-atomic, we still have $\mathbb{P}[\mathcal{E}] \ge \frac{1}{2}$. We now construct a partition of $\mathcal{X}$ as follows. Let $(x^k)_k$ be a dense sequence of $\mathcal{X}$. We denote by $B(x,r) = \{x' \in \mathcal{X}, \rho(x,x') < r\}$ the ball centered at x of radius $r > 0$. For any $k \ge 1$ and $\delta > 0$ let $P_k(\delta) = B(x^k, \delta) \setminus \bigcup_{l<k} B(x^l, \delta)$. Then, $(P_k(\delta))_k$ forms a partition of $\mathcal{X}$. For any $\delta > 0$ and sequence $\boldsymbol{b} = (b_k)_{k\ge1}$ in $\{0,1\}$ we

consider the following deterministic rewards

$$r_{\delta,\boldsymbol{b}}(a \mid x) = \begin{cases} b_k & a = a_1,\ x \in P_k(\delta), \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\}. \end{cases}$$

Now for any sequence of binary sequences $\mathbf{b} = (\boldsymbol{b^i})_{i\geq 0}$ where $\boldsymbol{b^i} = (b^i_k)_{k\geq 1}$, we will consider the memoryless rewards $\boldsymbol{r^b}$ defined as follows. The deterministic rewards r_t being constructed for $t \leq T_{k-1}$, we pose $r^{\mathbf{b}}_t = r_t$ for $t \leq T_{k-1}$. For all idle phases, i.e., $T_{k-1} < t < T^0$ or $2T^i \leq T^{i+1}$ for $i \geq 0$, we pose $r^{\mathbf{b}}_t = 0$. Last, for any $i \geq 0$ and $T^i \leq t < 2T^i$ we pose $r^{\mathbf{b}}_t = r_{\delta_i,\boldsymbol{b^i}}$. Now let $\mathbf{b}$ be a random sequence such that all $\boldsymbol{b^i}$ are independent i.i.d. Bernoulli $\mathcal{B}(\frac{1}{2})$ sequences in $\{0,1\}$. On the event $\mathcal{E}$, all new instances fall in distinct sets of the partitions defining the rewards. Hence, with this perspective, the reward of the action a_2 is always $\frac{3}{4}$ while on the event $\mathcal{E}$, for each new instance value, the reward of a_1 is a random Bernoulli $\mathcal{B}(\frac{1}{2})$. Intuitively, for a specific instance x, if the learner has not yet explored the arm a_1, selecting a_1 incurs an average regret $\frac{1}{4}$ compared to selecting the fixed arm a_2. We will then argue that there is a time T_k and a realization of $\tilde{\mathbb{X}}_{\leq T_k}$ and rewards, such that on this realization, the regret compared to the best actions for each instance in hindsight is significantly large. We now formalize these ideas.

Because $\tilde{\mathbb{X}}$ is a CS process, there exists a universally consistent learning rule under $\tilde{\mathbb{X}}$. Then, because $f.$ is optimistically universal, it is universally consistent under $\tilde{\mathbb{X}}$. Now fix a specific realization of the sequences in $\mathbf{b}$, considering the policy which always plays action a_2, i.e. $\pi_0 : x \in \mathcal{X} \mapsto a_2 \in \mathcal{A}$, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r^{\mathbf{b}}_t(a_2 \mid X_t) - r^{\mathbf{b}}_t(\hat{a}_t \mid X_t) \leq 0, \quad (a.s.).$$

In particular, since $\mathbb{P}[\mathcal{E}] \geq \frac{1}{2}$, we have

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r^{\mathbf{b}}_t(a_2 \mid X_t) - r^{\mathbf{b}}_t(\hat{a}_t \mid X_t) \mid \mathcal{E}, \mathbf{b}\right] \leq 0.$$

As a result, taking the expectation over $\mathbf{b}$ then applying Fatou's lemma gives

$$\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} r^{\mathbf{b}}_t(a_2 \mid X_t) - r^{\mathbf{b}}_t(\hat{a}_t \mid X_t) \mid \mathcal{E}\right] \leq 0.$$

Now let $\alpha_k := \frac{1}{16\cdot 4^{1/\epsilon_k}}$. In particular, there exists $i \geq \frac{4}{\alpha_k}$ such that for all $T \geq T^i$,

$$\mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} r^{\mathbf{b}}_t(a_2 \mid X_t) - r^{\mathbf{b}}_t(\hat{a}_t \mid X_t) \mid \mathcal{E}\right] \leq \frac{\alpha_k}{4}. \tag{6.1}$$

For simplicity, we may write $r^{\mathbf{b}}_t(a)$ instead of $r^{\mathbf{b}}_t(a \mid x)$, when it is clear from context that $x = X_t$. We now focus on period $[T^i, 2T^i)$ and denote by $\mathcal{S}^i_p := \{T^i + (p-1)\cdot\epsilon_k T^i \leq t <$

$T^i + p \cdot \epsilon_k T^i\}$ the sub-phase p for $1 \le p \le \frac{1}{\epsilon_k}$ of this period. Also note by A_p^i the number of new exploration steps for arm a_1 during $\mathcal{S}_p^i$, i.e., times when the learner selected a_1 for an instance that had not previously been explored

$$\mathcal{A}_p^i = \{t \in \mathcal{S}_p^i : \hat{a}_t = a_1, \forall 1 \le q < p : \hat{a}_{t+(q-p)\epsilon_k T^i} \neq a_1\}, \quad A_p^i = |\mathcal{A}_p^i|.$$

We show by induction that $\mathbb{E}[A_p^i \mid \mathcal{E}] \le 4^{p+1}\alpha_k T^i$ for all $1 \le p \le \frac{1}{\epsilon_k}$. Let $1 \le p \le \frac{1}{\epsilon_k}$. Suppose that the result was shown for $1 \le q < p$ (if $p = 1$ this is directly satisfied). We have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T^i(1+p\epsilon_k)-1} r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \\
&\ge -2T^{i-1} + \mathbb{E}\left[\sum_{t=T^i(1+(p-1)\epsilon_k)}^{T^i(1+p\epsilon_k)-1} (r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t))\mathbb{1}_{\mathcal{A}_p^i}(t) - \sum_{q<p} \frac{(p+1-q)A_q^i}{4} \mid \mathcal{E}\right] \\
&= -2T^{i-1} - \sum_{q<p} \frac{p+1-q}{4}\mathbb{E}[A_q^i \mid \mathcal{E}] \\
&\qquad + \mathbb{E}\left[\sum_{t=T^i(1+(p-1)\epsilon_k)}^{T^i(1+p\epsilon_k)-1} \mathbb{1}_{\mathcal{A}_p^i}(t)\mathbb{E}[r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) | t \in \mathcal{A}_p^i, \mathcal{E}] \,\middle|\, \mathcal{E}\right]
\end{aligned}$$

where in the first inequality we discard times from phase $\mathcal{S}_p^i$ for which an exploration of the corresponding instance during phases $\mathcal{S}_1^i, \dots \mathcal{S}_{p-1}^i$: these yield a regret least $(3/4-1) = -1/4$ compared to the fixed arm a_2. For each instance newly explored during phase $\mathcal{S}_q^i$, i.e. $t \in \mathcal{S}_q^i$, it affects potentially the $(p+1-q)$ next times with the same instance in phases $\mathcal{S}_q^i, \dots, \mathcal{S}_p^i$. Now, note that all elements in $\mathbf{b}$ are together independent, and independent from the process $\mathbb{X}$, in particular, independent from $\mathcal{E}$. As a result, the rewards at a time $\mathcal{A}_p^i$ are independent of the past because X_t visits a set of the partition $(P_k(\delta_i))_k$ which has never been visited. Thus, we have

$$\mathbb{E}[r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) | t \in \mathcal{A}_p, \mathcal{E}] = \frac{3}{4} - \frac{0+1}{2} = \frac{1}{4}.$$

Combining the above estimates with Eq (6.1) then gives

$$\begin{aligned}
-2T^{i-1} - \frac{1}{4}\sum_{q<p}(p+1-q)\mathbb{E}[A_q^i \mid \mathcal{E}] + \frac{1}{4}\mathbb{E}[A_p^i \mid \mathcal{E}] &\le \mathbb{E}\left[\sum_{t=1}^{T^i(1+p\epsilon_k)-1} r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \\
&\le \frac{\alpha_k}{4}T^i(1+p\epsilon_k) \le \frac{\alpha_k}{2}T^i.
\end{aligned}$$

Thus,

$$\begin{aligned}\mathbb{E}[A_p^i \mid \mathcal{E}] &\leq \left(\frac{8}{1+i} + 2\alpha_k\right) T^i + \sum_{q<p}(p+1-q)\mathbb{E}[A_q^i \mid \mathcal{E}] \\ &\leq 4\alpha_k T^i \left(1 + \sum_{q=1}^{p-1}(p+1-q)4^q\right) \\ &\leq 4\alpha_k T^i \left(1 + \sum_{q=1}^{p-1} 2^{p-q}4^q\right) = 4\alpha_k T^i \left(1 + 2^p(2^p-1)\right) \leq 4^{p+1}\alpha_k T^i.\end{aligned}$$

This completes the induction.

For any time t, denote $a_t^* = \arg\max_{a\in\mathcal{A}} r_t^{\mathbf{b}}(a)$ the optimal arm in hindsight. Note that $a_t^* \in \{a_1, a_2\}$. We lower bound the regret of the learner compared to the best action in hindsight until time T^{i+1}. To do so, define $\mathcal{B} = \bigcup_{p=1}^{1/\epsilon_k}\{t \in \mathcal{S}_p^i : \forall 1 \leq q \leq p, t + (q-p)\epsilon_k T^i \notin \mathcal{A}_q^i\}$ the set of times t such that the learner never explored a_1 on the present and past appearances of the instance X_t. We also define $\mathcal{C} = \{T^i \leq t < 2T^i : a_t^* = a_1\}$ the set of times when a_1 was the optimal action. One can observe that for any time in $\mathcal{B}$, because no exploration on a_1 was performed up for the corresponding instance X_t in the past history, $\mathbb{P}[t \in \mathcal{C} | t \in \mathcal{B}, \mathcal{E}] = \frac{1}{2}$. Hence, if $t \in \mathcal{B}\cap\mathcal{C}\cap\mathcal{E}$, the learner incurs a regret at least $\frac{1}{4}$ compared to the best arm $a_t^* = a_1$. Therefore,

$$\mathbb{E}\left[\sum_{t=1}^{2T^i-1} r_t^{\mathbf{b}}(a_t^*) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \geq \frac{1}{4}\mathbb{E}\left[\sum_{t\in\mathcal{B}} \mathbb{1}_{\mathcal{C}}(t) \mid \mathcal{E}\right] = \frac{1}{8}\mathbb{E}[|\mathcal{B}| \mid \mathcal{E}].$$

where by construction, we have $|\mathcal{B}| + \sum_{p=1}^{1/\epsilon_k}\left(\frac{1}{\epsilon_k} - p + 1\right) A_p^i = 2T^i - T^i = T^i$. As a result,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{2T^i-1} r_t^{\mathbf{b}}(a_t^*) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] &\geq \frac{T^i}{8} - \frac{\alpha_k}{2}T^i \sum_{p=1}^{1/\epsilon_k}\left(\frac{1}{\epsilon_k} - p + 1\right)4^p \\ &\geq \frac{T^i}{8} - \alpha_k T^i 4^{1/\epsilon_k} \\ &\geq \frac{T^i}{16} \geq \frac{2T^i - 1}{32}.\end{aligned}$$

Hence, there exists a realization of instances $\boldsymbol{X}_{<2T^i} \leq \tilde{\mathbb{X}}_{<2T^i}$ falling in $\mathcal{E}$ and of rewards $(r_t)_{<2T^i}$ such that the regret compared to the best action in hindsight for on this specific instance sequence and for these rewards is at least $\frac{T^i}{16}$. We then pose $T_k := 2T^i - 1$, and use the realization $\boldsymbol{X}_{\leq T_k}$, $(r_t)_{\leq T_k}$ for the deterministic process $\mathbb{X}_{\leq T_k}$ and $(r_t)_{t\leq T_k}$. We recall that by construction, the realizations are consistent with the previously constructed process $\mathbb{X}_{\leq T_{k-1}}$ and rewards $(r_t)_{\leq T_{k-1}}$. Further, to each new instance between times T^i and $2T^i - 1$ corresponded the best action in hindsight: this gives a collection of pairs (x, a) where $x \in \mathcal{X}$ is an instance visited by the deterministic process $\mathbb{X}$ between times T^i and $2T^i - 1$ and $a \in \{a_1, a_2\}$ is the corresponding best action. Let $\mathcal{D}_k$ denote this collection. This ends the recursive construction of the deterministic process $\mathbb{X}$ and rewards.

Because we enforced that the samples of μ be always distinct and different from $\bar{x}$ across the construction of $\mathbb{X}$, the countable collection $\bigcup_{k\geq 1} \mathcal{D}_k$ of pairs instance/optimal-action never contains pairs with the same instance x. Hence, we can consider the following measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$ defined by

$$\pi^*(x) = \begin{cases} a & \text{if } (x,a) \in \bigcup_{k\geq 1} \mathcal{D}_k, \\ a_2 & \text{otherwise.} \end{cases}$$

This policy always performs the optimal action in hindsight. Hence by construction, for any $k \geq 1$,

$$\mathbb{E}\left[\frac{1}{T_k}\sum_{t=1}^{T_k} r_t(\pi^*(X_t) \mid X_t) - r_t(\hat{a}_t \mid X_t)\right] \geq \frac{1}{32},$$

where $\hat{a}_t$ refers to the learner's decisions on the constructed process $\mathbb{X}$ and rewards $(r_t)_{t\geq 1}$. Note that the expectation is taken only with respect to the learner's randomness given that $\mathbb{X}$ and $(r_t)_{t\geq 1}$ are deterministic. Because the above equation holds for all $k \geq 1$ and $(T_k)_{k\geq 1}$ is an increasing sequence of times, we have

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t)\right] \geq \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t)\right] \geq \frac{1}{32},$$

where we used Fatou's lemma. This proves that $f_\cdot$ is not universally consistent on $\mathbb{X}$.

We now show that $\mathbb{X} \in \text{SMV}$. It suffices to check that it visits a sublinear number of distinct points—this is also necessary since $\mathbb{X}$ is deterministic. For $t \geq 1$, denote by N_t the number of distinct instances visited by the process $\mathbb{X}_{\leq t}$. Fix $k \geq 1$. The process $\mathbb{X}_{\leq T_k}$ being constructed from the process $\tilde{\mathbb{X}}_{\leq T_k}$ above, we re-use the same notations. Let $i \geq 1$ such that $T_k = 2T^i - 1$. For $1 \leq j \leq i$ and $T^j \leq t < \min(T^{j+1}, T_k)$ we have $N_t \leq T_{k-1} + 1 + n_j + k_j \leq 1 + \epsilon_k T^0 + 2k_j \leq 1 + 3\epsilon_k T^j \leq 1 + 3\epsilon_k t$. (The additional 1 accounts for $\bar{x}$.) For $T_{k-1} < t < T^0$, we have $N_t \leq 1 + N_{t_{k-1}} \leq 2 + 3\epsilon_{k-1} t$. As a result for all $T_{k-1} < t \leq T_k$ we have

$$N_t \leq 2 + 3\epsilon_{k-1} t.$$

Because $\epsilon_k \to 0$ as $k \to \infty$, we obtain that $\frac{N_t}{t} \to 0$ as $t \to \infty$. This shows that $\mathbb{X} \in \text{SMV}$. Because $\mathbb{X}$ is deterministic and in SMV, Proposition 6.4 shows that there exists a universally consistent learning rule on $\mathbb{X}$. However, $f_\cdot$ is not universally consistent under $\mathbb{X}$ which contradicts the hypothesis. This ends the proof that there does not exist an optimistically universal learning rule. ■

We now turn to the case of spaces $\mathcal{X}$ which do not have a non-atomic measure and show that in this case, the learning rule for processes visiting a sublinear number of distinct instances in Proposition 6.3 is an optimistically universal learning rule for all settings including online rewards.

Theorem 6.5. *Let $\mathcal{X}$ be a metrizable separable Borel space such that there does not exist a non-atomic probability measure on $\mathcal{X}$, and $\mathcal{A}$ a finite action space. Then, learnable processes are exactly $\text{SOAB}_{stat} = \text{SOAB}_{online} = \text{SMV}$ and there exists an optimistically universal learning rule for all settings.*

Proof We show that any process $\mathbb{X} \in \text{SMV}$ visits a sublinear number of distinct instances almost surely. Fix $\mathbb{X} \in \text{SMV}$. Using Lemma 5.3 from Chapter 5, because $\mathcal{X}$ does not admit a non-atomic probability measure, there exists a countable set $Supp(\mathbb{X})$ such that on an event $\mathcal{E}$ of probability one, for all $t \geq 1$, $X_t \in Supp(\mathbb{X})$. Then consider the sequence $(\{x\})_{x \in Supp(\mathbb{X})}$ of disjoint measurable sets of $\mathcal{X}$. Applying the SMV property of $\mathbb{X}$ to this sequence yields $|\{x \in Supp(\mathbb{X}) : \{x\} \cap \mathbb{X}_{\leq T}\}| = o(T)$, $(a.s.)$. We denote by $\mathcal{F}$ the corresponding event of probability one. By union bound $\mathbb{P}[\mathcal{E} \cap \mathcal{F}] = 1$. Now on the event $\mathcal{E}$, for any $T \geq 1$ we have

$$|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = |\{x \in Supp(\mathbb{X}) : \{x\} \cap \mathbb{X}_{\leq T}\}|.$$

As a result, on the event $\mathcal{E} \cap \mathcal{F}$ we have $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$, which proves the claim that SMV visit a sublinear number of distinct instances almost surely. As a result, the learning rule $f.$ from Proposition 6.3 which simply performs independent copies of the EXP3.IX algorithm for each distinct visited instance is universally consistent under all processes $\mathbb{X} \in \text{SMV}$. Now recall that in the stationary case, the condition SMV is already necessary for universal learning. This condition is already necessary for universal learning in the noiseless full-feedback setting [Han21a]. As a result, $\text{SOAB}_{online} \subset \text{SOAB}_{stat} = \text{SMV}$. Therefore, universally learnable processes are exactly SMV even in the online rewards setting and $f.$ is optimistically universal, which completes the proof. ■

6.5 Universally Learnable Processes for Context Spaces with Non-Atomic Probability Measures

6.5.1 Necessary conditions on learnable processes

In the previous section, we showed that for spaces $\mathcal{X}$ that do not have non-atomic probability measures, the set of learnable processes is exactly SMV, independently of the learning setting. Here, we focus on the remaining case of universal learning for spaces $\mathcal{X}$ that admit a non-atomic probability measure for adversarial rewards and aim to understand which processes admit universal learning. We focus here on necessary conditions; sufficient conditions are given in the next section.

Condition $\mathcal{C}_4$ is necessary for universal learning with oblivious rewards

We quickly recall the definition of condition $\mathcal{C}_4$. For an integer $i \geq 0$ and any $k \geq 1$, we define $T_i^k = \lfloor 2^u(1 + v2^{-i}) \rfloor$ where $k = u2^i + v$ and $u \geq 0, 0 \leq v < 2^i$ are integers. In particular, $u = \lfloor k2^{-i} \rfloor$ and $v = k \bmod 2^i$. These times form periods $[T_i^k, T_i^{k+1})$ which become finer as i increases. Then consider the set of times t such that X_t is the first appearance of the instance on its period,

$$\mathcal{T}^i = \{t \geq 1 : T_i^k \leq t < T_i^{k+1},\ \forall T_i^k \leq t' < t, X_{t'} \neq X_t\}.$$

We note that the sets $\mathcal{T}^p$ are increasing with p. We recall that $\mathcal{C}_4$ is defined as follows.

Condition 4. *For any sequence of disjoint measurable sets $(A_i)_{i\geq 1}$ of $\mathcal{X}$, we have*

$$\lim_{i\to\infty} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{A_i}(X_t)\right] = 0.$$

Denote by $\mathcal{C}_4$ the set of all processes $\mathbb{X}$ satisfying this condition.

We first give an alternative definition of $\mathcal{C}_4$ which will be useful in the next results.

Proposition 6.5. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathbb{X}$ a stochastic process on $\mathcal{X}$. The following are equivalent.*

- $\mathbb{X} \in \mathcal{C}_4$,
- *For any sequence of decreasing measurable sets $(A_i)_{i\geq 1}$ with $A_i \downarrow \emptyset$,*

$$\sup_{p\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_i}(X_t)\right] \underset{i\to\infty}{\longrightarrow} 0.$$

- *For any sequence of decreasing measurable sets $(A_i)_{i\geq 1}$ with $A_i \downarrow \emptyset$,*

$$\mathbb{E}\left[\sup_{p\geq 0} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_i}(X_t)\right] \underset{i\to\infty}{\longrightarrow} 0.$$

Proof Suppose that the second proposition is not satisfied. We aim to show that $\mathbb{X} \notin \mathcal{C}_4$. By hypothesis, there exists measurable sets $A_i \downarrow \emptyset$, $\epsilon > 0$, and an increasing sequence of indices $(i_p)_{p\geq 1}$ such that

$$\sup_{l\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^l} \mathbb{1}_{A_{i_p}}(X_t)\right] \geq \epsilon.$$

Now let $i \geq 1$ and $p \geq 1$ such that $i_p \geq i$. We observe that because $A_{i_p} \subset A_i$,

$$\sup_{l\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^l} \mathbb{1}_{A_i}(X_t)\right] \geq \sup_{l\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^l} \mathbb{1}_{A_{i_p}}(X_t)\right] \geq \epsilon.$$

Hence, for any $i \geq 1$, there exists $p(i) > 0$ such that

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{p(i)}} \mathbb{1}_{A_i}(X_t)\right] \geq \frac{\epsilon}{2}.$$

Case 1. We consider a first case where there exists $\eta_i > 0$ such that for any $j \geq i$,

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{p(i)}} \mathbb{1}_{A_j}(X_t)\right] \geq \eta_i.$$

For simplicity, we will write $T^k = T^k_{p(i)}$. We will also drop the indices i of $p(i)$ and η_i for conciseness. We now construct by induction a sequence of indices $(k(l))_{l\geq 0}$ together with indices $(j(l))_{l\geq 0}$ with $k(0) = 1$, $j(0) = i$ and such that for any $l \geq 1$,

$$\mathbb{E}\left[\sup_{T^{k(l-1)}<T\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}\setminus A_{j(l)}}(X_t)\right] \geq \frac{\eta}{2}.$$

Suppose that we have already constructed $j(0), \ldots, j(l-1)$ and $k(0), \ldots, k(l-1)$. Note that

$$\mathbb{E}\left[\sup_{T>T^{k(l-1)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}}(X_t)\right] \geq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}}(X_t)\right] \geq \eta.$$

Therefore, by the dominated convergence theorem, there exists $k(l) > k(l-1)$ such that

$$\mathbb{E}\left[\sup_{T^{k(l-1)}<t\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}}(X_t)\right] \geq \frac{3\eta}{4}.$$

Now because $A_i \downarrow \emptyset$, there exists $j(l) > j(l-1)$ such that $\mathbb{P}[A_{j(l)} \cap \mathbb{X}_{\leq T^{k(l)}} = \emptyset] \geq 1 - \frac{\eta}{4}$. Let us denote by $\mathcal{E}$ this event. Then,

$$\begin{aligned}
&\mathbb{E}\left[\sup_{T^{k(l-1)}<t\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}\setminus A_{j(l)}}(X_t)\right] \\
&\geq \mathbb{E}\left[\mathbb{1}[\mathcal{E}] \sup_{T^{k(l-1)}<t\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}\setminus A_{j(l)}}(X_t)\right] \\
&= \mathbb{E}\left[\mathbb{1}[\mathcal{E}] \sup_{T^{k(l-1)}<t\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}}(X_t)\right] \\
&\geq \mathbb{E}\left[\sup_{T^{k(l-1)}<t\leq T^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_{j(l-1)}}(X_t)\right] - \frac{\eta}{4} \geq \frac{\eta}{2}.
\end{aligned}$$

This ends the construction of the indices $k(l)$ and $j(l)$ for $l \geq 1$. Now for any $u \geq 1$, let $S_u = \{l \geq 1 : l \equiv 2^{u-1} \bmod 2^u\}$. The main remark is that S_u is infinite for all $u \geq 1$ and they are all disjoint. We then pose $B_u = \bigcup_{l\in S_u} A_{j(l-1)} \setminus A_{j(l)}$. Because all S_u are disjoint, this implies that the sets $(B_u)_u$ are also disjoint. Then, using Fatou's lemma together with

the fact that all S_u are infinite, we obtain

$$\begin{aligned}\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^p}\mathbb{1}_{B_u}(X_t)\right] &\geq \limsup_{k\in S_u}\mathbb{E}\left[\sup_{T^{k(l-1)}<T\leq T^{k(l)}}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^p}\mathbb{1}_{B_u}(X_t)\right]\\ &\geq \limsup_{k\in S_u}\mathbb{E}\left[\sup_{T^{k(l-1)}<T\leq T^{k(l)}}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^p}\mathbb{1}_{A_{j(l-1)}\setminus A_{j(l)}}(X_t)\right]\\ &\geq \frac{\eta}{2}.\end{aligned}$$

We obtain therefore for any $u\geq p$

$$\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^u}\mathbb{1}_{B_u}(X_t)\right]\geq\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^p}\mathbb{1}_{B_u}(X_t)\right]\geq\frac{\eta}{2}.$$

This ends the proof that $\mathbb{X}\notin\mathcal{C}_4$.

Case 2. Recalling that the sets $(A_i)_i$ are decreasing, we can now suppose that for all $i\geq 1$, one has $\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i)}}\mathbb{1}_{A_j}(X_t)\right]\to 0$ as $j\to\infty$. We now construct a sequence of indices $(i(u))_{u\geq 1}$ as follows such that $i(1)=1$ and for any $u\geq 1$,

$$\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u))}}\mathbb{1}_{A_{i(u)}\setminus A_{i(u+1)}}(X_t)\right]\geq\frac{\epsilon}{4}.$$

Suppose we have constructed $i(u)$. Then, by the hypothesis of this case, there exists $i(u{+}1)>i(u)$ such that

$$\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u))}}\mathbb{1}_{A_{i(u+1)}}(X_t)\right]\leq\frac{\epsilon}{4}.$$

Now note that

$$\begin{aligned}\frac{\epsilon}{2}\leq\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u))}}\mathbb{1}_{A_{i(u)}}(X_t)\right] &\leq \mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u))}}\mathbb{1}_{A_{i(u)}\setminus A_{i(u+1)}}(X_t)\right]\\ &\quad+\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u))}}\mathbb{1}_{A_{i(u+1)}}(X_t)\right].\end{aligned}$$

As a result, the induction at step p is complete. We then define a sequence of measurable sets $(B_j)_{j\geq 1}$ such that for any $u\geq 1$, $B_{p(i(u))}=A_{i(u)}-A_{i(u+1)}$, and for all other indices $j\notin\{p(i(u)),u\geq 1\}$ we set $B_j=\emptyset$. All these sets are disjoint, and we have for any $u\geq 1$,

$$\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T,t\in\mathcal{T}^{p(i(u)}}\mathbb{1}_{B_{p(i(u))}}(X_t)\right]\geq\frac{\epsilon}{4}.$$

Therefore, $\mathbb{X} \notin \mathcal{C}_4$.

We now show that if $\mathbb{X}$ satisfies the second property, then $\mathbb{X} \in \mathcal{C}_4$. Let $(A_i)_i$ be a sequence of disjoint measurable sets, and define $B_i = \bigcup_{j \geq i} A_j$. Then,

$$0 \leq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{A_i}(X_t)\right] \leq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{B_i}(X_t)\right]$$
$$\leq \sup_{p\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{B_i}(X_t)\right].$$

Hence, since $B_i \downarrow \emptyset$, the second property shows that $\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{A_i}(X_t)\right] \to 0$ as $i \to \infty$.

Now for any Borel set A, by the dominated convergence theorem and the fact that the sets $\mathcal{T}^p$ are increasing for $p \geq 0$, we obtain

$$\lim_{p\to\infty} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A}(X_t)\right] = \mathbb{E}\left[\lim_{p\to\infty} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A}(X_t)\right],$$

where both terms are bounded by 1. In other terms,

$$\sup_{p\geq 0} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A}(X_t)\right] = \mathbb{E}\left[\sup_{p\geq 0} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A}(X_t)\right].$$

As a result, the second and third conditions of the proposition are equivalent. ■

The main result of this section is that the $\mathcal{C}_4$ condition is necessary for universal learning with oblivious rewards.

Theorem 6.6. *Let $\mathcal{X}$ be a metrizable separable Borel space, and a finite action space $\mathcal{A}$ with $|\mathcal{A}| \geq 2$. Then, $\text{SOAB}_{\text{oblivious}} \subset \mathcal{C}_4$.*

Proof Fix, $a_1, a_2 \in \mathcal{A}$ two distinct actions. By contradiction, let $\mathbb{X} \notin \mathcal{C}_4$ and $f.$ a universally consistent learning rule under $\mathbb{X}$ for oblivious rewards. For simplicity, we will denote by $\hat{a}_t$ the action selected by the learning rule at time t. By hypothesis, let $(A_i)_{i\geq 1}$ be a sequence of disjoint measurable sets and $0 < \epsilon \leq 1$ such that

$$\limsup_{i\to\infty} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{A_i}(X_t)\right] \geq \epsilon.$$

Then, there exists an increasing sequence $(j(i))_{i\geq 1}$ such that for any $p \geq 1$,

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{j(i)}} \mathbb{1}_{A_{j(i)}}(X_t)\right] \geq \frac{\epsilon}{2}.$$

We write $\mathcal{I} = \{j(i), i \geq 1\}$. Without loss of generality, we can suppose $A_j = \emptyset$ if $j \in \mathcal{I}$. We now construct recursively rewards $(r_t)_{t\geq 1}$ on which this algorithm is not consistent, as well as a policy $\pi^* : \mathcal{X} \to \mathcal{A}$ compared to which the algorithm has high regret. The reward functions and policy are constructed recursively together with an increasing sequence of times $(T^p)_{p\in\mathcal{I}}$ such that after the $p-$th iteration of the construction process, the rewards r_t for $t \leq T^p$ have been defined such that $r_t(\cdot \mid x_{\leq t}) = 0$ if $x \notin \bigcup_{i<p} A_i$, the policy $\pi^*(\cdot)$ is defined on $\bigcup_{i<p} A_i$ and always the best action in hindsight until T^{p-1}. For $p = j(p')$, suppose that we have performed $p'-1$ iterations of this construction and have constructed the times $T^{j(1)}, \dots, T^{j(p'-1)}$. For convenience, let $\alpha_p = 2^{-p-1}$ and define $K_p = \left\lceil \frac{2}{\alpha_p} \log \frac{2^6}{\epsilon} \right\rceil$, $\beta_p = \frac{\epsilon}{2^{10}(1+2\alpha_p)^{(K_p-1)K_p} 4^{K_p}}$, $\tilde{K}_p = \left\lceil \frac{2}{\alpha_p} \log \frac{8}{\beta_p} \right\rceil$ and $M_p = \max(\frac{8}{\epsilon\alpha_p}, (1+2\alpha_p)^{K_p+\tilde{K}_p})$. We first construct by induction an increasing sequence of indices $(k(l))_{l\geq 0}$ with $k(0) = \min\{k \geq 2^p : T_p^k > M_p T^{j(p'-1)}\}$ and such that for any $l \geq 1$, $T_p^{k(l)} > M_p T_p^{k(l-1)}$ and

$$\mathbb{E}\left[\max_{M_p T_p^{k(l-1)} < T \leq T_p^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] \geq \frac{\epsilon}{4}.$$

To do so, suppose that we have constructed $k(l')$ for $0 \leq l' < l$. Note that

$$\mathbb{E}\left[\sup_{T > M_p T_p^{k(l-1)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] \geq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] \geq \frac{\epsilon}{2}.$$

Then, by dominated convergence theorem, there exists $k(l) > k(l-1)$ such that $T_p^{k(l)} > M_p T_p^{k(l-1)}$ and

$$\mathbb{E}\left[\max_{M_p T_p^{k(l-1)} < T \leq T_p^{k(l)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] \geq \mathbb{E}\left[\sup_{T > M_p T_p^{k(l-1)}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] - \frac{\epsilon}{4} \geq \frac{\epsilon}{4}.$$

This ends the construction of the sequence $(k(l))_{l\geq 0}$. We then denote by $\hat{k}(l)$ the index of a phase $(T_p^{k-1}, T_p^k]$ where the max is attained, i.e.

$$\hat{k}(l) = \operatorname*{arg\,max}_{k\leq k(l)} \left(\max_{M_p T_p^{k(l-1)}, T_p^{k-1} < T \leq T_p^k} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t) \right).$$

Ties can be broken in alphabetical order. Because $T_p^k \leq 2T_p^{k-1}$, we have in particular,

$$\mathbb{E}\left[\frac{1}{T_p^{\hat{k}(l)}} \sum_{t\leq T_p^{\hat{k}(l)}, t\in\mathcal{T}^p} \mathbb{1}_{A_p}(X_t)\right] \geq \frac{\epsilon}{8}.$$

Now for any $l \geq 1$, let δ_l such that

$$\mathbb{P}\left[\min_{1\leq t,t'\leq T_p^{k(l)}, X_t \neq X_{t'}} \rho(X_t, X_{t'}) \leq \delta_l\right] \leq \frac{\epsilon}{2^{l+10}}.$$

Then, let $\mathcal{E}$ be the event when for all $l \geq 1$, we have $\min_{1 \leq t,t' \leq T_p^{k(l)}, X_t \neq X_{t'}} \rho(X_t, X_{t'}) > \delta_l$. By the union bound, $\mathbb{P}[\mathcal{E}] \geq 1 - \frac{\epsilon}{2^{10}}$. As a result, we have

$$\mathbb{E}\left[\frac{1}{T_p^{\hat{k}(l)}} \sum_{t \leq T_p^{\hat{k}(l)}, t \in \mathcal{T}^p} \mathbb{1}_{A_p}(X_t) \mid \mathcal{E}\right] \geq \frac{\epsilon}{16}. \tag{6.2}$$

Now for $\delta > 0$ and $u \geq 1$, define the sets $P_u(\delta) = (A_p \cap B(x^u, \delta)) \setminus \bigcup_{v<u} B(x^v, \delta)$ which form a partition of A_p. For any $\delta > 0$ and sequence $\boldsymbol{b} = (b_u)_{u \geq 1}$ in $\{0,1\}$ we consider the following deterministic rewards

$$r_{\delta,\boldsymbol{b}}(a \mid x) = \begin{cases} b_u & a = a_1,\ x \in P_u(\delta), \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\}, \end{cases} \qquad \text{if } x \in A_p, \qquad r_{\delta,\boldsymbol{b}}(\cdot \mid x) = 0 \text{ if } x \notin A_p.$$

For any sequence of binary sequences $\mathbf{b} = (\boldsymbol{b^k})_{k \geq 0}$ where $\boldsymbol{b^k} = (b_u^k)_{u \geq 1}$, and binary sequence $\boldsymbol{c} = (c_k)_{k \geq 0}$ we construct the rewards $\boldsymbol{r}^{\mathbf{b},\boldsymbol{c}}$ as follows. For $t \leq T^{j(p'-1)}$ we pose $r_t^{\mathbf{b},\boldsymbol{c}} = r_t$ so that the rewards $\boldsymbol{r}^{\mathbf{b},\boldsymbol{c}}$ coincide with those constructed by induction so far. For $T^{j(p'-1)} < t \leq T_p^{k(0)}$ we pose $r_t^{\mathbf{b},\boldsymbol{c}} = 0$. For $t > T_p^{k(0)}$ let $l \geq 1$ such that $T_p^{k(l-1)} < t \leq T_p^{k(l)}$ and $k > k(0)$ such that $T_p^{k-1} < t \leq T_p^k$. Then, we pose

$$r_t^{\mathbf{b},\boldsymbol{c}}(a \mid x_{\leq t}) = \begin{cases} 0 & \exists t' \leq T_p^{k(l-1)} : x_{t'} = x_t \\ 0 & \text{o.w. } c_k = 0, \\ r_{\delta_l,\boldsymbol{b^l}}(a \mid x_t) & \text{o.w. } c_k = 1, \forall T_p^{k-1} < t' < t : x_{t'} \neq x_t, \\ 0 & \text{o.w. } c_k = 1, \exists T_p^{k-1} < t' < t : x_{t'} = x_t, \end{cases}$$

for $a \in \mathcal{A}, x_{\leq t} \in \mathcal{X}^t$. Note that these rewards coincide with the rewards that have been constructed by induction so far. Now let $\mathbf{b}$ be generated such that all $\boldsymbol{b^k}$ are independent i.i.d. Bernoulli $\mathcal{B}(\frac{1}{2})$ random sequences in $\{0,1\}$, and $\boldsymbol{c}$ is also an independent i.i.d. $\mathcal{B}(\frac{1}{2})$ process. The sequence is used to delete some periods $(T_p^{k-1}, T_p^k]$. Precisely, for any $l \geq 1$, we consider the following event where we deleted the periods between $\hat{k}(l) - K_p - \tilde{K}_p$ and $\hat{k}(l) - K_p$ but did not delete periods after this phase until period $\hat{k}(l)$,

$$\mathcal{F}_l^p = \bigcap_{\hat{k}(l)-K_p-\tilde{K}_p < k \leq \hat{k}(l)-K_p} \{c_k = 0\} \cap \bigcap_{\hat{k}(l)-K_p < k \leq \hat{k}(l)} \{c_k = 1\}.$$

One can note that the events $\mathcal{F}_l^p$ for $l \geq 1$ are together independent. Indeed, $\hat{k}(l) \leq k(l)$ and $T_p^{\hat{k}(l)} > M_p T^{k(l-1)} \geq (1+2\alpha_p)^{K_p+\tilde{K}_p} T^{k(l-1)}$, which yields $\hat{k}(l) > k(l-1) + K_p + \tilde{K}_p$. As a result, the indices of $\boldsymbol{c}$ considered in the events $\mathcal{F}^p$ all lie in distinct intervals $(k(l-1), k(l)]$, hence their independence. Further, we have $\mathbb{P}[\mathcal{F}_l^p] = 2^{-K_p-\tilde{K}_p}$. Then, the Borel-Cantelli implies that on an event $\mathcal{F}^p$ of probability one, there is an infinite number of $l \geq 1$ such that $\mathcal{F}_l^p$ is satisfied.

Next, define $\pi_0 : x \in \mathcal{X} \mapsto a_2 \in \mathcal{A}$, the policy which always selects arm a_2. Fix any realization of $\mathbf{b}$ and $\boldsymbol{c}$. Because $f.$ is universally consistent for oblivious rewards, it

has in particular sublinear regret compared to π_0 under rewards $\boldsymbol{r}^{\mathbf{b},\boldsymbol{c}}$, i.e., almost surely $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2 \mid X_t) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t \mid X_t)) \leq 0$. Now observe that the event $\mathcal{F}^p$ only depends on $\boldsymbol{c}$ and $\mathbb{X}$ and is in particular independent from $\mathbf{b}$. Therefore, $\mathbb{P}[\mathcal{E}\cap\mathcal{F}^p \mid \mathbf{b}] = \mathbb{P}[\mathcal{E}\cap\mathcal{F}^p] \geq 1 - \frac{\epsilon}{2^{10}}$, where we used $\mathbb{P}[\mathcal{F}^p] = 1$. Therefore,

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2 \mid \mathbb{X}_{\leq t}) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t \mid \mathbb{X}_{\leq t}) \mid \mathcal{E}, \mathcal{F}^p, \mathbf{b}\right] \leq 0.$$

For conciseness, we will omit the terms $\mathbb{X}_{\leq t}$ in the rest of the proof. We then take the expectation over $\mathbf{b}$ and $\boldsymbol{c}$. Thus, by the dominated convergence theorem, there exists $l_0 \geq 1$ such that

$$\mathbb{E}\left[\sup_{T>T_p^{k(l_0)}} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}, \mathcal{F}^p\right] \leq \frac{\beta_p}{8}.$$

On the event $\mathcal{F}^p$, there exists $\hat{l} > l_0$ such that the event $\mathcal{F}^p_{\hat{l}}$ is met. For convenience, we take $\hat{l}$ as the minimum index satisfying these conditions. Then, we have

$$\begin{aligned}
&\mathbb{E}\left[\sup_{T_p^{\hat{k}(\hat{l})-K_p}<T\leq T_p^{\hat{k}(\hat{l})}} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}, \mathcal{F}^p\right] \\
&\leq \mathbb{E}\left[\sup_{T_p^{k(\hat{l}-1)}<T\leq T_p^{k(\hat{l})}} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}, \mathcal{F}^p\right] \leq \frac{\beta_p}{8}.
\end{aligned}$$

Now let l^p such that $\mathbb{P}[\hat{l} \leq l^p \mid \mathcal{F}^p] \geq \frac{1}{2}$. Then,

$$\mathbb{E}\left[\sup_{T_p^{\hat{k}(\hat{l})-K_p}<T\leq T_p^{\hat{k}(\hat{l})}} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_2) - r_t^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}, \mathcal{F}^p, \hat{l} \leq l_p\right] \leq \frac{\beta_p}{4}. \tag{6.3}$$

For conciseness, we will write $\hat{k}$ for $\hat{k}(\hat{l})$, let $\mathcal{G}^p = \mathcal{E}\cap\mathcal{F}^p\cap\{\hat{l}\leq l_p\}$. We now use similar arguments as in the proof of Theorem 6.4, to show that the learning rule incurs a large regret compared to the best action in hindsight, before time $T_p^{\hat{k}}$. We focus on the period $(T_p^{\hat{k}-K_p}, T_p^{\hat{k}}]$, which we decompose using the sets

$$\mathcal{S}_q = \{T_p^{\hat{k}-K_p-1+q} < t \leq T_p^{\hat{k}-K_p+q} : X_t \in A_p\} \cap \mathcal{T}^p, \quad 1 \leq q \leq K_p.$$

We also define E_q the number of new exploration steps for arm a_1 during $\mathcal{S}_q$,

$$\begin{aligned}
&Exp_q := \\
&\left\{t \in \mathcal{S}_q : \hat{a}_t = a_1 \text{ and } \forall t' \in \bigcup_{q'<q} \mathcal{S}_{q'} : X_{t'} = X_t,\ \hat{a}_{t'} \neq a_1\right\} \setminus \{t : \exists t' \leq T_p^{\hat{k}-\tilde{K}_p}, X_{t'} = X_t\},
\end{aligned}$$

and $E_q = |Exp_q|$. We now show by induction on i that $\mathbb{E}\left[\frac{E_q}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right] \leq (1+2\alpha_p)^{(q-1)K_p} 4^{q+1}\beta_p$ for all $1 \leq q \leq K_p$. Suppose that this is shown for all $1 < q' < q$. Recalling that on the event $\mathcal{G}^p$, for any $T_p^{\hat{k}-K_p-\tilde{K}_p} < t \leq T_p^{\hat{k}-K_p}$ we have $r_t^{\mathbf{b},\mathbf{c}} = 0$, we can use the same arguments as in Theorem 6.4 to obtain

$$
\begin{aligned}
&\mathbb{E}\left[\frac{1}{T_p^{\hat{k}-K_p+q}} \sum_{t=1}^{T_p^{\hat{k}-K_p+q}} r_t^{\mathbf{b},\mathbf{c}}(a_2) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] \\
&\geq -\mathbb{E}\left[\frac{T_p^{\hat{k}-K_p-\tilde{K}_p}}{T_p^{\hat{k}-K_p+q}} \mid \mathcal{G}^p\right] + \sum_{q'=1}^{q} \mathbb{E}\left[\frac{1}{T_p^{\hat{k}-K_p+q}} \sum_{t=T_p^{\hat{k}-K_p-1+q'}+1}^{T_p^{\hat{k}-K_p+q'}} r_t^{\mathbf{b},\mathbf{c}}(a_2) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] \\
&= -\mathbb{E}\left[\frac{T_p^{\hat{k}-K_p-\tilde{K}_p}}{T_p^{\hat{k}-K_p+q}} \mid \mathcal{G}^p\right] + \sum_{q'=1}^{q} \mathbb{E}\left[\frac{1}{T_p^{\hat{k}-K_p+q}} \sum_{t\in\mathcal{S}^{q'}} r_t^{\mathbf{b},\mathbf{c}}(a_2) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] \\
&\geq -(1+q)\mathbb{E}\left[\frac{T_p^{\hat{k}-K_p-\tilde{K}_p}}{T_p^{\hat{k}-K_p+q}} \mid \mathcal{G}^p\right] - \sum_{q'<q} \frac{q+1-q'}{4}\mathbb{E}\left[\frac{E_{q'}}{T_p^{\hat{k}-K_p}} \mid \mathcal{G}^p\right] \\
&\qquad + \mathbb{E}\left[\frac{1}{T_p^{\hat{k}}} \sum_{t=T_p^{\hat{k}-K_p-1+q}+1}^{T_p^{\hat{k}-K_p+q}} \mathbb{1}_{Exp_q}(t)(r_t^{\mathbf{b},\mathbf{c}}(a_2) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t)) \mid \mathcal{G}^p\right],
\end{aligned}
$$

where the additional terms $-T_p^{\hat{k}-\tilde{K}_p}$ compared to the computations in Theorem 6.4 are due to the fact that in Exp_q we also discard times of instances that were visited before $T^{\hat{k}-\tilde{K}_p}$, and that in a single period $\mathcal{S}_q$, there are no duplicates. Now for any $T^{\hat{k}-K_p-1+q} < t \leq T^{\hat{k}-K_p+q}$ such that a pure exploration was performed $t \in Exp_q$, we have

$$
\mathbb{E}[r_t^{\mathbf{b},\mathbf{c}}(a_2) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid t \in Exp_q, \mathcal{G}^p, \hat{k}] = \frac{3}{4} - \frac{0+1}{2} = \frac{1}{4},
$$

because X_t visits a set of the partition $(P_u(\delta_{\hat{k}-K_p+q}))_u$ which has never been visited in the past, hence, the reward of a_1 on this set is equally likely to be 0 or 1 (depending on $\mathbf{b}$), and $\mathcal{G}^p$ is independent of $\mathbf{b}$. Also, using the inequality $\log(1+z) \geq \frac{z}{2}$ for $0 \leq z \leq 1$ we obtain $T_p^{\hat{k}-\tilde{K}_p} \leq (1+\alpha_p)^{-\tilde{K}_p}(1+T_p^{\hat{k}-K_p}) \leq \frac{\beta_p}{8}(1+T_p^{\hat{k}-K_p}) \leq \frac{\beta_p}{4}T_p^{\hat{k}-K_p}$. Lastly, $T_p^{\hat{k}-K_p} \geq T_p^{\hat{k}}/(1+2\alpha_p)^{K_p}$. Combining these results with Eq (6.3) yields

$$
\frac{\beta_p}{4} \geq -(1+q)\frac{\beta_p}{4} - \frac{1}{4}\sum_{q'<q}(q+1-q')(1+2\alpha_p)^{K_p}\mathbb{E}\left[\frac{E_{q'}}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right] + \frac{1}{4}\mathbb{E}\left[\frac{E_q}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right].
$$

Thus,

$$\begin{aligned}\mathbb{E}\left[\frac{E_q}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right] &\leq (2+q)\beta_p + (1+2\alpha_p)^{K_p}\sum_{q'<q}(q+1-q')\mathbb{E}\left[\frac{E_{q'}}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right] \\ &\leq (1+2\alpha_p)^{(q-1)K_p}\beta_p\left(2+q+4\sum_{q'=1}^{q-1}(q+1-q')4^{q'}\right) \\ &\leq (1+2\alpha_p)^{(q-1)K_p}4^{q+1}\beta_p.\end{aligned}$$

This completes the induction. Now for any $t \geq 1$, denote by $a_t^* = \arg\max_{a\in\mathcal{A}} r_t^{\mathbf{b},\mathbf{c}}(a \mid \mathbb{X}_{\leq t})$ the optimal action in hindsight. In particular, $a_t^* \in \{a_1, a_2\}$. Now define

$$\mathcal{B} = \bigcup_{q=1}^{K_0}\left\{t \in \mathcal{S}_q : \forall t' \in \bigcup_{q'<q}\mathcal{S}_{q'} : X_{t'} = X_t, t \notin Exp_{q'}\right\}.$$

These are times such that we never explored the action a_2. In particular, on $\mathcal{G}^p$, the learner incurs an average regret of at least $\frac{1}{8}$ on these times since action a_2 would be optimal with probability $\frac{1}{2}$ with a reward excess $\frac{1}{4}$ over action a_1. Therefore,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T_p^{\hat{k}}}\sum_{t=1}^{T_p^{\hat{k}}} r_t^{\mathbf{b},\mathbf{c}}(a_t^*) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] &\geq \mathbb{E}\left[\frac{1}{T_p^{\hat{k}}}\sum_{T_p^{\hat{k}-K_p}<t\leq T_p^{\hat{k}}} r_t^{\mathbf{b},\mathbf{c}}(a_t^*) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] \\ &\geq \frac{1}{8}\mathbb{E}\left[\frac{|\mathcal{B}|}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right].\end{aligned}$$

Now denote by $T_p^* = |\{t \leq T_p^{\hat{k}} : X_t \in A_p\} \cap \mathcal{T}^p|$. Recall that because $\mathcal{F}^p$ and $\hat{l}$ are independent from $\mathcal{E}$, by Eq (6.2), we have $\mathbb{E}\left[\frac{T_p^*}{T_p^{\hat{k}}} \mid \mathcal{G}^p\right] = \left[\frac{T_p^*}{T_p^{\hat{k}}} \mid \mathcal{E}\right] \geq \frac{\epsilon}{16}$. By construction, we have $|\mathcal{B}| + \sum_{q=1}^{K_p}(K_p - q + 1)E_q + K_p T_p^{\hat{k}-K_p-\tilde{K}_p} \geq T_p^* - T_p^{\hat{k}-K_p}$. Thus,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T_p^{\hat{k}}}\sum_{t=1}^{T_p^{\hat{k}}} r_t^{\mathbf{b},\mathbf{c}}(a_t^*) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] &\geq \frac{\epsilon}{2^7} - \frac{K_p}{4}\mathbb{E}\left[\frac{T_p^{\hat{k}-K_p-\tilde{K}_p}}{T_p^{\hat{k}}}\right] \\ &\quad - \frac{\beta_p}{2}\sum_{q=1}^{K_p}(K_p-q+1)(1+2\alpha_p)^{(q-1)K_p}4^q \\ &\geq \frac{\epsilon}{2^7} - \frac{\beta_p K_p}{16} - \frac{\beta_p}{2}(1+2\alpha_p)^{(K_p-1)K_p}4^{K_p+1} \\ &\geq \frac{\epsilon}{2^8}.\end{aligned}$$

Recall that by construction $\mathbb{P}[\hat{l} \leq l^p \mid \mathcal{F}^p] \geq \frac{1}{2}$. Also, $\mathbb{P}[\mathcal{F}^p] = 1$ and both these events are independent from $\mathcal{E}$, hence , letting $T^p = T_p^{k(l^p)}$ we have

$$\mathbb{E}\left[\sup_{T^{p-1}<T\leq T^p}\frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\mathbf{c}}(a_t^*) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{E}\right] \geq \frac{1}{2}\mathbb{E}\left[\frac{1}{T_p^{\hat{k}}}\sum_{t=1}^{T_p^{\hat{k}}} r_t^{\mathbf{b},\mathbf{c}}(a_t^*) - r_t^{\mathbf{b},\mathbf{c}}(\hat{a}_t) \mid \mathcal{G}^p\right] \geq \frac{\epsilon}{2^9}.$$

This ends the construction of the sequence T^p. Then, for any binary sequences $\boldsymbol{b}$ and $\boldsymbol{c}$ we introduce slightly different rewards $(\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}})_{t\le T^p}$ as follows: for $t \le T^{j(p'-1)}$, $\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}} = r_t$, for $T^{j(p'-1)} < t \le T_p^{k(0)}$ let $\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}} = 0$. For $t > T_p^{k(0)}$ let $l \ge 1$ such that $T_p^{k(l-1)} < t \le T_p^{k(l)}$ and $k > k(0)$ such that $T_p^{k-1} < t \le T_p^k$. Then, we pose

$$\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}(a \mid x_{\le t}) = \begin{cases} 0 & \exists t' \le T_p^{k(l-1)} : x_{t'} = x_t \\ 0 & \text{o.w. } c_k = 0, \\ r_{\delta_{l^p},\boldsymbol{b}}(a \mid x_t) & \text{o.w. } c_k = 1, \forall T_p^{k-1} < t' < t : x_{t'} \neq x_t, \\ 0 & \text{o.w. } c_k = 1, \exists T_p^{k-1} < t' < t : x_{t'} = x_t, \end{cases}$$

for $a \in \mathcal{A}, x_{\le t} \in \mathcal{X}^t$. The only difference with the previous oblivious rewards is that we use the same reward function $r_{\delta_{l^p},\boldsymbol{b}}$ across phases $(T_p^{k(l-1)}, T_p^{k(l)}]$ for $l \le l^p$. Then, consider the following policy,

$$\pi^{\boldsymbol{b}}(x) = \begin{cases} a_1 & \text{if } b_u = 1, x \in P_u(\delta_{l^p}) \cap A_p, \\ a_2 & \text{if } b_u = 0, x \in P_u(\delta_{l^p}) \cap A_p \\ \pi^*(x) & \text{if } x \in \bigcup_{i<p} A_i \\ a_1 & \text{if } x \notin \bigcup_{i\le p} A_i. \end{cases}$$

Note that by induction hypothesis on the rewards r_t for $t \le T^{j(p'-1)}$, using the rewards $\tilde{\boldsymbol{r}}^{\boldsymbol{b},\boldsymbol{r}}$, $\pi^{\boldsymbol{b}}$ always selects the best action in hindsight for times $t \le T^{j(p'-1)}$. Also, by construction, $\pi^{\boldsymbol{b}}$ also selects the best action in hindsight for times $T^{j(p'-1)} < t \le T^p$.

Similarly to before, suppose that $\boldsymbol{b}, \boldsymbol{c}$ are generated as independent i.i.d. $\mathcal{B}(\frac{1}{2})$ processes. We now argue that on the event $\mathcal{E}$, the learning process with rewards $\boldsymbol{r}^{\mathbf{b},\boldsymbol{c}}$ until T^p is stochastically equivalent to the learning process with rewards $\tilde{\boldsymbol{r}}^{\boldsymbol{b},\boldsymbol{c}}$ until T^p. Indeed, these rewards only differ in that for different periods $(T_p^{k(l-1)}, T_p^{k(l)}]$, we may have reward $r_{\delta_l,\boldsymbol{b}^{\boldsymbol{l}}}$ instead of $r_{\delta_{l^p},\boldsymbol{b}}$. However, on the event $\mathcal{E}$, new instances always fall in portions where the reward of a_1 is still $\mathcal{B}(\frac{1}{2})$ conditionally on the currently available history. This holds for both reward sequences. Further, duplicates can only affect rewards during the same period $(T_p^{k(l-1)}, T_p^{k(l)}]$ by construction—if x_t is a duplicate from a previous period, the reward function is 0. Hence, even though for $\boldsymbol{r}^{\mathbf{b},\boldsymbol{c}}$, we have distinct sequences $\boldsymbol{b}^{\boldsymbol{l}}$, these are all consistent with a single sequence $\boldsymbol{b}$ based on a finer partition at scale δ_{l^p}. Precisely, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{b},\boldsymbol{c}}\left[\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sup_{T^{j(p'-1)}<T\le T^p} \frac{1}{T}\sum_{t=1}^{T} \tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}(\pi^{\boldsymbol{b}}(X_t)) - \tilde{r}^{\boldsymbol{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}\right]\right] \\
&\quad = \mathbb{E}_{\mathbb{X}}\left[\mathbb{E}_{\boldsymbol{b},\boldsymbol{c}}\mathbb{E}_{\hat{a}}\left[\sup_{T^{j(p'-1)}<T\le T^p} \frac{1}{T}\sum_{t=1}^{T} \tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}(a_t^*)) - \tilde{r}^{\boldsymbol{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathbb{X}, \mathcal{E}\right] \mid \mathcal{E}\right] \\
&\quad = \mathbb{E}_{\mathbb{X}}\left[\mathbb{E}_{\mathbf{b},\boldsymbol{c}}\mathbb{E}_{\hat{a}}\left[\sup_{T^{j(p'-1)}<T\le T^p} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_t^*) - r^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathbb{X}, \mathcal{E}\right] \mid \mathcal{E}\right] \\
&\quad = \mathbb{E}\left[\sup_{T^{j(p'-1)}<T\le T^p} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b},\boldsymbol{c}}(a_t^*) - r^{\mathbf{b},\boldsymbol{c}}(\hat{a}_t) \mid \mathcal{E}\right] \ge \frac{\epsilon}{2^9}.
\end{aligned}$$

As a result, there exists a specific realization of $\boldsymbol{b}$ and $\boldsymbol{c}$ such that

$$\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sup_{T^{j(p'-1)}<T\leq T^p}\frac{1}{T}\sum_{t=1}^{T}\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}(\pi^{\boldsymbol{b}}(X_t))-\tilde{r}^{\boldsymbol{b},\boldsymbol{c}}(\hat{a}_t)\mid\mathcal{E}\right]\geq\frac{\epsilon}{2^9}.$$

Hence, because $\mathbb{P}[\mathcal{E}^c]\leq\frac{\epsilon}{2^{10}}$, we obtain

$$\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sup_{T^{j(p'-1)}<T\leq T^p}\frac{1}{T}\sum_{t=1}^{T}\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}(\pi^{\boldsymbol{b}}(X_t))-\tilde{r}^{\boldsymbol{b},\boldsymbol{c}}(\hat{a}_t)\right]\geq\frac{\epsilon}{2^9}\left(1-\frac{\epsilon}{2^{10}}\right)-\frac{\epsilon}{2^{10}}\geq\frac{\epsilon}{2^{11}}.$$

Now for all $t\leq T^p$ we pose $r_t=\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}$, and complete the definition of π^* by setting $\pi^*(x)=\pi^{\boldsymbol{b}}(x)$ on $\bigcup_{i\leq p}A_i$. Note that these definitions are consistent with the previously constructed rewards and the actions selected by the policy on $\bigcup_{i<p}A_i$. This ends the recursive construction of the rewards $\boldsymbol{r}=(r_t)_{t\geq 1}$ and the policy π^* on $\bigcup_{i\geq 1}A_i$. We close the definition of π^* by setting $\pi^*(x)=a_1$ for $x\notin\bigcup_{i\geq 1}A_i$ arbitrarily. The constructed policy π^* is measurable because it is measurable on each A_i for $i\geq 1$.

We now analyze the regret of the algorithm compared to π^* for the rewards $(r_t)_t$. First, note that the rewards are deterministic and that π^* is the optimal policy, i.e., which always selects the best arm in hindsight. Also, if $\boldsymbol{b},\boldsymbol{c}$ denote the realizations used in the iteration $p=j(p')$ of the above recursion, for any $t\leq T^p$ we have $r_t=\tilde{r}_t^{\boldsymbol{b},\boldsymbol{c}}$. As a result,

$$\mathbb{E}\left[\sup_{T^{j(p'-1)}<T\leq T^{j(p')}}\frac{1}{T}\sum_{t=1}^{T}r_t(\pi^b(X_t))-r_t(\hat{a}_t)\right]\geq\frac{\epsilon}{2^{11}}.$$

Now by Fatou's lemma, we have

$$\begin{aligned}
&\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}r_t(\pi^b(X_t))-r_t(\hat{a}_t)\right]\\
&=\mathbb{E}\left[\limsup_{p'\to\infty}\sup_{T^{j(p'-1)}<T\leq T^{j(p')}}\frac{1}{T}\sum_{t=1}^{T}r_t(\pi^b(X_t))-r_t(\hat{a}_t)\right]\\
&\geq\limsup_{p\to\infty}\mathbb{E}\left[\sup_{T^{j(p'-1)}<T\leq T^{j(p')}}\frac{1}{T}\sum_{t=1}^{T}r_t(\pi^b(X_t))-r_t(\hat{a}_t)\right]\\
&\geq\frac{\epsilon}{2^{11}}.
\end{aligned}$$

As a result, $f.$ is not consistent on the oblivious rewards $(r_t)_t$ under $\mathbb{X}$, which contradicts the hypothesis that $f.$ is universally consistent under $\mathbb{X}$. This ends the proof. ■

Recall that the condition SMV is necessary for universal learning because this is already the case for noiseless online learning [Han21a] and is also sufficient for universal learning in noiseless online learning (Chapter 3), online learning with adversarial responses (Chapter 4) and stationary contextual bandits (Chapter 5). In the next proposition, we show that our new necessary condition $\mathcal{C}_4$ is stronger than SMV.

Proposition 6.6. *Let $\mathcal{X}$ be a metrizable separable Borel space. Then, $\mathcal{C}_4 \subset$ SMV.*

Proof Suppose that $\mathbb{X} \notin$ SMV, then there exists a sequence of disjoint sets $(A_i)_{i\geq 1}$ and $\epsilon > 0$ such that $\mathbb{E}[\limsup_{T\to\infty} \frac{1}{T}|\{i \geq 1, A_i \cap \mathbb{X}_{\leq T} \neq \emptyset\}|] \geq \epsilon$. We now let $B_i = \bigcup_{j\geq i} A_j$. We define $\bar{\mathcal{T}} = \{t \geq 1 : \forall t' < t, X_{t'} \neq X_t\}$ the set of new instances times. Then, for any $i \geq 1$,

$$\begin{aligned}
\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^i} \mathbb{1}_{B_i}(X_t)\right] &\geq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\bar{\mathcal{T}}} \mathbb{1}_{B_i}(X_t)\right] \\
&\geq \mathbb{E}\left[\limsup_{T\to\infty} \frac{|\{j \geq i : A_j \cap \mathbb{X}_{\leq T} \neq \emptyset\}|}{T}\right] \\
&= \mathbb{E}\left[\limsup_{T\to\infty} \frac{|\{j \geq 1 : A_j \cap \mathbb{X}_{\leq T} \neq \emptyset\}|}{T}\right] \\
&\geq \epsilon.
\end{aligned}$$

This holds for all $i \geq 1$ and $B_i \downarrow \emptyset$. Hence, the second property of Proposition 6.5 implies $\mathbb{X} \notin \mathcal{C}_4$. ■

Further, $\mathcal{C}_4$ is a strictly stronger condition than SMV provided that $\mathcal{X}$ admits a non-atomic probability measure. More precisely, in the next result, we explicitly construct a process $\mathbb{X} \in$ SMV $\setminus \mathcal{C}_4$ which does not admit universal learning even in the memoryless setting. As a result, for memoryless, oblivious, prescient, and online rewards, one cannot universally learn all SMV processes, while this was achievable for stationary rewards. Thus having adversarial partial feedback on the losses of each action strictly reduces the set of learnable processes $\text{SOAB}_{online} \subset \text{SOAB}_{oblivious} \subset \text{SOAB}_{memoryless} \subsetneq$ SMV.

Theorem 6.7. *Let $\mathcal{X}$ be a metrizable separable Borel space such that there exists a non-atomic probability measure on $\mathcal{X}$, and a finite action space $\mathcal{A}$ with $|\mathcal{A}| \geq 2$. Then, $\mathcal{C}_4 \subsetneq$ SMV and the set of learnable processes also satisfies $\text{SOAB}_{memoryless} \subsetneq$ SMV.*

Before proving this result, we present a lemma that allows to have a countable sequence of non-atomic measures with disjoint support.

Lemma 6.2. *Let $\mathcal{X}$ be a metrizable separable Borel space such that there exists a non-atomic probability measure on $\mathcal{X}$. Then, there exists a sequence of disjoint non-empty measurable sets $(A_i)_{i\geq 0}$ and probability measures $(\nu_i)_{i\geq 0}$ on $\mathcal{X}$ such that $\nu_i(A_i) = 1$.*

Proof Let ρ denote the metric on $\mathcal{X}$. First, let $(x^i)_{i\geq 1}$ be a dense sequence on $\mathcal{X}$. For any $x \in \mathcal{X}$ and $r > 0$ we denote by $B(x, r) = \{x' \in \mathcal{X} : \rho(x, x') < \delta\}$ the open ball centered at x of radius r. Then, for any $\delta > 0$, we define the partition $\mathcal{P}(\delta) = (P_i(\delta))_{i\geq 1}$ by $P_i(\delta) = B(x^i, \delta) \setminus \bigcup_{j<i} B(x^j, \delta)$.

Let μ_{-1} a non-atomic probability measure on $\mathcal{X}$. We construct the disjoint measures and sets recursively. We pose $B_0 = \mathcal{X}$. Suppose for $p \geq 1$ that we have constructed disjoint sets $(A_i)_{i\leq p-1}$, disjoint with B_{p-1}, as well as non-atomic probability measures $(\nu_i)_{i\leq p-1}$ and μ_{p-1} satisfying $\nu_i(A_i) = 1$ for $i \leq p-1$ and $\mu_{p-1}(B_{p-1}) = 1$. Now let $Z_1, Z_2 \sim \mu_{p-1}$ two independent random variables with distribution μ_{p-1}. Because μ_{p-1} is non-atomic, $Z_1 \neq Z_2$ almost surely. Thus, there exists $\delta_p > 0$ such that $\mathbb{P}[\rho(Z_1, Z_2) \leq \delta_p] \leq \frac{1}{2}$. As a result,

with probability at least $\frac{1}{2}$, Z_1 and Z_2 fall in distinct sets of the partition $\mathcal{P}(\delta_p)$. Hence, there exists at least two indices $i < j$ such that $\mathbb{P}[Z_1 \in P_i(\delta_p)], \mathbb{P}[Z_2 \in P_j(\delta_p)] > 0$. We then pose $A_p = B_{p-1} \cap P_i(\delta_p)$ and $B_p = B_{p-1} \cap P_j(\delta_p)$. Because $\mu_{p-1}(B_{p-1}) = 1$, we have $\mu_{p-1}(A_p) = \mu_{p-1}(P_i(\delta_p)) > 0$. Similarly, $\mu_{p-1}(B_p) > 0$. Hence, we can consider the probability measure ν_p of μ_{p-1} conditionally on A_p (i.e. $\nu_p(A) = \frac{\mu_{p-1}(A \cap A_p)}{\mu_{p-1}(A_p)}$ for all measurable A). Similarly, let μ_p the probability measure of μ_{p-1} conditionally on B_p. Both are non-atomic because the original measure μ_{p-1} is non-atomic. This ends the recursion and the proof of the lemma. ■ We are now ready to prove the theorem.

Proof of Theorem 6.7 Fix $a_1, a_2 \in \mathcal{A}$ two distinct actions. Let $(x^i)_{i \geq 1}$ be a dense sequence of $\mathcal{X}$ and denote by $B(x, r)$ denotes the open ball centered at $x \in \mathcal{X}$ with radius $r > 0$. Using Lemma 6.2, let $(A_i)_{i \geq 0}$ disjoint measurable sets together with non-atomic probability measures $(\nu_i)_{i \geq 0}$ such that $\nu_i(A_i) = 1$. We then fix $x_0 \in A_0$ (we will not use the set A_0 any further and from now will only reason on the sets $(A_i)_{i \geq 1}$) and for $i \geq 1$, we define $S_i = \{k \geq 1 : k \equiv 2^{i-1} \bmod 2^i\}$. Then let $\mathbb{Z}^i$ for $i \geq 1$ be independent processes where $\mathbb{Z}^i$ is an i.i.d. process following the distribution ν_i. We now construct a process $\mathbb{X}$ on $\mathcal{X}$. For any $k \geq 1$, let $T_k = 2^k k!$, $n_i = 2^{\lfloor \log_2 i \rfloor}$ for $i \geq 1$, and $l_k = \sum_{l \in S_i, l < k} \frac{T_k}{n_i}$, where $k \equiv 2^{i-1} \bmod 2^i$. For any $t \geq 1$, we pose

$$X_t = \begin{cases} Z^i_{l_k + r} & \text{if } T_k \leq t < 2T_k, k \equiv 2^{i-1} \bmod 2^i, t - T_k \equiv r \bmod \frac{T_k}{n_i}, 1 \leq r \leq \frac{T_k}{n_i}, \\ x_0 & \text{otherwise.} \end{cases}$$

This ends the construction of $\mathbb{X}$. We now argue that $\mathbb{X} \in \text{SMV}$. Let $(B_l)_{l \geq 1}$ be a sequence of disjoint measurable sets of $\mathcal{X}$. Because $\mathbb{Z}^i$ is an i.i.d. process for any $i \geq 1$, the event $\mathcal{E}_i$ where $|\{l : \mathbb{Z}^i_{\leq T} \cap B_l \neq \emptyset\}| = o(T)$ has probability one. Now define $\mathcal{E} = \bigcap_{i \geq 1} \mathcal{E}_i$, which has probability one by the union bound. Fix $\epsilon > 0$ and $i^* = \lceil \frac{2}{\epsilon} \rceil$ so that $\epsilon \leq \frac{1}{n_{i^*}}$. On the event $\mathcal{E}$ for any $i \leq i^*$ there exists T_i such that for any $T \geq T_i$ we have $|\{l : \mathbb{Z}^i_{\leq T} \cap B_l \neq \emptyset\}| \leq \frac{\epsilon}{2^i} T$. Now let $T^0 = \max_{i \leq i^*} T_i n_i$. Then, on $\mathcal{E}$, for any $T \geq T^0$,

$$\begin{aligned} |\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}| &\leq 1 + \sum_{i=1}^{i^*} |\{l : \mathbb{Z}^i_{\leq \lfloor T/n_i \rfloor} \cap B_l \neq \emptyset\}| \\ &\quad + |\{l : \exists t \leq T : X_t \in B_l, T_k \leq t < 2T_k, k \equiv 0 \bmod 2^{i^*}\}| \\ &\leq 1 + \epsilon T + |\{X_t, \quad t \leq T, T_k \leq t < 2T_k, k \equiv 0 \bmod 2^{i^*}\}| \\ &\leq 1 + \epsilon T + \frac{T}{n_{i^*}} + \frac{T}{n_{i^*}} \\ &\leq 3\epsilon T + 1. \end{aligned}$$

In the first inequality, the additional 1 is due to the visit of x_0, and in the third inequality, we used the fact that in a phase $i > i^*$, each point is duplicated $n_i \geq n_{i^*}$ times. This yields a term $\frac{T}{n_{i^*}}$. The second term $\frac{T}{n_{i^*}}$ in the third inequality is due to boundary effects for times close to T, the worst-case scenarios being attained for T of the form $T_k(1 + \frac{1}{n_i})$. As a result, on $\mathcal{E}$, we have $\limsup_{T \to \infty} \frac{1}{T} |\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}| \leq 3\epsilon$, which holds for any $\epsilon > 0$. Thus, $\frac{1}{T} |\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}| \to 0$ on $\mathcal{E}$, which ends the proof that $\mathbb{X} \in \text{SMV}$.

We now show that there does not exist a universally consistent algorithm under $\mathbb{X}$ for memoryless rewards. One can easily check that $\mathbb{X} \notin \mathcal{C}_4$, since for any $i \geq 1$, we have

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{\lfloor \log_2 i\rfloor}} \mathbb{1}_{A_i}(X_t)\right] \geq \mathbb{E}\left[\limsup_{k\to\infty} \frac{\mathbb{1}_{S_i}(k)}{2T_k} \sum_{t\leq 2T_k, t\in\mathcal{T}^{\lfloor \log_2 i\rfloor}} \mathbb{1}_{A_i}(X_t)\right]$$
$$\geq \mathbb{E}\left[\limsup_{k\to\infty} \frac{\mathbb{1}_{S_i}(k)}{2}\right] \geq \frac{1}{2}.$$

This already shows that $\textsc{Soab}_{online} \subset \textsc{Soab}_{oblivious} \subset \mathcal{C}_4 \subsetneq \text{SMV}$. However, we will show a stronger statement that $\mathbb{X} \notin \textsc{Soab}_{memoryless}$. The proof uses the same techniques as Theorem 6.6, but leverages the fact that the phases S^i are deterministic and instances from previous phases $[T_k, 2T_k)$ do not appear in future phases. By contradiction, suppose that $f_\cdot$ is a universally consistent learning rule. We will refer to its decision at time t as $\hat{a}_t$ for simplicity. We will construct recursively rewards $(r_t)_{t\geq 1}$ on which this algorithm is not consistent, as well as a policy $\pi^* : \mathcal{X} \to \mathcal{A}$ compared to which the algorithm has high regret. The rewards and policy are constructed recursively together with an increasing sequence of times $(T^p)_{p\geq 1}$ and indices $(i_p)_{p\geq 1}$ with $i_1 = 1$ such that after the $p-$th iteration of the construction process, the rewards $r_t(a \mid \cdot)$ have been defined for all $t \leq T^p$ and the policy $\pi^*(\cdot)$ has been defined $\bigcup_{i<i_p} A_i$. The rewards will be deterministic and stationary, hence we may omit the subscript t. Suppose that we have performed $p-1$ iterations of this construction for $p \geq 1$. We will drop the subscripts p for simplicity and simply assume that we have defined the reward $r(a \mid \cdot)$ and the value of the policy $\pi^*(\cdot)$ on $\bigcup_{j<i} A_j$ for some $i \geq 1$ ($i = i_p$). We now construct the rewards on A_i. To do so, we will first introduce other memoryless rewards. For any $k \in S_i$, because ν_i is non-atomic, there exists δ_k such that

$$\mathbb{P}\left[\min_{1\leq u<v\leq l_k+\frac{T_k}{n_i}} \rho(Z_u^i, Z_v^i) \leq \delta_k\right] \leq 2^{-k-5}.$$

Then, let $\mathcal{E}^i$ be the event when for all $k \in S_i$, we have $\min_{1\leq u<v\leq l_k+\frac{T_k}{n_i}} \rho(Z_u^i, Z_v^i) > \delta_k$, and $\mathbb{Z}^i$ takes values in A_i only—this is almost sure since $\nu_i(A_i) = 1$. By the union bound, $\mathbb{P}[\mathcal{E}^i] \geq 1-\frac{1}{32}$. Now for $\delta > 0$ and $u \geq 1$, define the sets $P_u(\delta) = (A_i \cap B(x^u, \delta)) \backslash \bigcup_{v<u} B(x^v, \delta)$ which form a partition of A_i. For any $\delta > 0$ and sequence $\boldsymbol{b} = (b_u)_{u\geq 1}$ in $\{0,1\}$ we consider the following deterministic rewards

$$r_{\delta,\boldsymbol{b}}(a \mid x) = \begin{cases} b_u & a = a_1,\ x \in P_u(\delta), \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\}, \end{cases} \quad \text{if } x \in A_i, \qquad r_{\delta,\boldsymbol{b}}(a \mid x) = r(a \mid x) \text{ if } x \in \bigcup_{j<i} A_j,$$

and $r_{\delta,\boldsymbol{b}}(\cdot \mid x) = 0$ if $x \notin \bigcup_{j\leq i}$. Now for any sequence of binary sequences $\mathbf{b} = (\boldsymbol{b}^k)_{k\in S_i}$ where $\boldsymbol{b}^k = (b_u^k)_{u\geq 1}$, we will consider the memoryless rewards $\boldsymbol{r}^{\mathbf{b}}$ defined as follows. For any $t \geq 2$, let $k \geq 1$ such that $T^k \leq t < T^{k+1}$, and $k' = \min\{l \in S_i : l \geq k\}$. We pose $r_t^{\mathbf{b}} = r_{\delta_{k'},\boldsymbol{b}^{k'}}$, and $r_1^{\mathbf{b}} = r_2^{\mathbf{b}}$. Now let $\mathbf{b}$ be generated such that all $\boldsymbol{b}^i$ are independent i.i.d. Bernoulli $\mathcal{B}(\frac{1}{2})$ random sequences in $\{0,1\}$. Next, define $\pi_0 : x \in \mathcal{X} \mapsto a_2 \in \mathcal{A}$, the policy which

always selects arm a_2. Now fix any realization of $\boldsymbol{r}^{\mathbf{b}}$. Because $f.$ is universally consistent for memoryless rewards, it has in particular sublinear regret compared to π_0 under rewards $\boldsymbol{r}^{\mathbf{b}}$, i.e., almost surely $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b}}(a_2 \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t)) \leq 0$. The same arguments as in Theorem 6.4 with Fatou's lemma give

$$\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b}}(a_2 \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right] \leq 0,$$

where the expectation is now also taken over $\mathbf{b}$. Therefore, with $\alpha_i := \frac{1}{16\cdot 4^{n_i}}$, there exists t_0 such that for all $T \geq t_0$, we have $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b}}(a_2 \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right] \leq \frac{\alpha_i}{4}$. In particular, there exists $k \in S_i$ such that $k \geq \frac{4}{\alpha_i}$ and $T_k \geq t_0$ and the above inequality holds for all $T_k \leq T < 2T_k$. Then, using the same arguments as in the proof of Theorem 6.4, if a_t^* denotes the best action in hindsight at time t, we have

$$\mathbb{E}\left[\sum_{t=T_k}^{2T_k-1} r_t^{\mathbf{b}}(a_t^* \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right] \geq \frac{T_k}{16}.$$

For any binary sequence $\boldsymbol{b}$, we will write for conciseness $r^{\boldsymbol{b}} = r_{\delta_k,\boldsymbol{b}}$. We also define the following policy, restricted to instances in A_i:

$$\pi^{\boldsymbol{b}} : x \in A_i \mapsto \begin{cases} a_1 & \text{if } b_u = 1, x \in P_u(\delta_k), \\ a_2 & \text{if } b_u = 0, x \in P_u(\delta_k). \end{cases}$$

Now consider the case where $\boldsymbol{b}$ is an i.i.d. sequence of Bernoullis $\mathcal{B}(\frac{1}{2})$. We argue that on the event $\mathcal{E}^i$, the learning process before time $2T_k - 1$ and under rewards $\boldsymbol{r}^{\mathbf{b}}$ is stochastically equivalent to the learning under stationary rewards $\boldsymbol{r}^{\boldsymbol{b}} := (r^{\boldsymbol{b}})_{t\geq 1}$ before $2T_k - 1$. Precisely, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{b}\sim\mathcal{B}(\frac{1}{2})}\left[\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sum_{t=T_k}^{2T_k-1} r^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t) \mid X_t) - r^{\boldsymbol{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right]\right] \\
&\quad = \mathbb{E}_{\mathbb{X}}\left[\mathbb{E}_{\boldsymbol{b}\sim\mathcal{B}(\frac{1}{2})}\mathbb{E}_{\hat{a}}\left[\sum_{t=T_k}^{2T_k-1} r^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t) \mid X_t) - r^{\boldsymbol{b}}(\hat{a}_t \mid X_t) \mid \mathbb{X}, \mathcal{E}^i\right] \mid \mathcal{E}^i\right] \\
&\quad = \mathbb{E}_{\mathbb{X}}\left[\mathbb{E}_{\mathbf{b}}\mathbb{E}_{\hat{a}}\left[\sum_{t=T_k}^{2T_k-1} r_t^{\mathbf{b}}(a_t^* \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t) \mid \mathbb{X}, \mathcal{E}^i\right] \mid \mathcal{E}^i\right] \\
&\quad = \mathbb{E}\left[\sum_{t=T_k}^{2T_k-1} r_t^{\mathbf{b}}(a_t^* \mid X_t) - r_t^{\mathbf{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right] \\
&\quad \geq \frac{T_k}{16},
\end{aligned}$$

where in the second inequality we used the fact that on the event $\mathcal{E}^i$, until time $2T_k - 1$ all distinct instances in A_i fall in distinct sets of the partition $(P_u(\delta_k))_u$: for both rewards $\boldsymbol{r}^{\boldsymbol{b}}$

and $\boldsymbol{r}^{\mathbf{b}}$, the reward on a new instance A_i is independent of the past and has the distribution $\mathcal{B}(\frac{1}{2})$ for action a_1 and deterministic $\frac{3}{4}$ for action a_2. As a result, there exists a specific realization of $\boldsymbol{b}$ such that

$$\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sum_{t=T_k}^{2T_k-1} r^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t) \mid X_t) - r^{\boldsymbol{b}}(\hat{a}_t \mid X_t) \mid \mathcal{E}^i\right] \geq \frac{T_k}{16}.$$

Hence, because $\mathbb{P}[(\mathcal{E}^i)^c] \leq \frac{1}{32}$, we obtain

$$\mathbb{E}_{\mathbb{X},\hat{a}}\left[\sum_{t=T_k}^{2T_k-1} r^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t) \mid X_t) - r^{\boldsymbol{b}}(\hat{a}_t \mid X_t)\right] \geq \frac{T_k}{16}\left(1 - \frac{1}{32}\right) - \frac{T_k}{32} \geq \frac{2T_k - 1}{2^7}.$$

Now denote $T^p = 2T_k - 1$, and let $i_{p+1} = 1 + \max\{j \geq i : \exists l \in S_i, T_l \leq T^i\} = 1 + \max\{j \geq i : T_{2^{j-1}} \leq T^i\}$. The index i_{p+1} is chosen so that until time T^p, the process $\mathbb{X}$ has not visited $\bigcup_{j \geq i_p} A_j$ yet. Note that this index is well defined since $T_k \to \infty$ as $k \to \infty$. We then pose $r(\cdot \mid x) = r^{\boldsymbol{b}}(\cdot \mid x)$ for all $x \in \bigcup_{i \leq j < i_{p+1}} A_j$. In particular, we have $r(a \mid x) = 0$ for all $x \in \bigcup_{i_p < j < i_{p+1}} A_j$. Then pose

$$\pi^*(x) = \begin{cases} \pi^{\boldsymbol{b}}(x) & x \in A_i \\ a_2 & x \in \bigcup_{i < j < i^{p+1}} A_j. \end{cases}$$

This ends the recursive construction of the reward r and the policy π^*, i.e., we have constructed $r(\cdot \mid x)$ and $\pi^*(x)$ for all $x \in \bigcup_{i \geq 1} A_i$. We end the definition of the rewards by posing $r_t(\cdot \mid x) = 0$ and $\pi^*(x) = a_2$ if $x \notin \bigcup_{i \geq 1} A_i$. Note that $(r_t)_{t \geq 1}$ forms a valid sequence of rewards since by construction on each A_i they are deterministic. Similarly, π^* is measurable because it is measurable on each A_i.

We now analyze the regret of the algorithm compared to π^* for the rewards $(r_t)_t$. First, note that the rewards are deterministic and time-independent, and that π^* is the optimal policy, i.e., which always selects the best arm in hindsight. Then, for any $p \geq 1$, we have

$$r(\cdot \mid x) = r^{\boldsymbol{b}}(\cdot \mid x), \quad \forall x \in \mathcal{X} \setminus \bigcup_{i \geq i_{p+1}} A_i.$$

where $r^{\boldsymbol{b}}$ denotes the rewards defined at the p-th iteration of the construction process. Now recall that by construction, the sets A_i visited by the process $\mathbb{X}_{\leq T^p}$ all satisfy $i < i_{p+1}$, which is the first index for which the rewards would differ. As a result, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{T^p}\sum_{t=1}^{T^p} r(\pi^*(X_t) \mid X_t) - r(\hat{a}_t \mid X_t)\right] &\geq \mathbb{E}\left[\frac{1}{T^p}\sum_{t=(T_p+1)/2}^{T^p} r(\pi^*(X_t) \mid X_t) - r(\hat{a}_t \mid X_t)\right] \\
&= \mathbb{E}\left[\frac{1}{T^p}\sum_{t=(T_p+1)/2}^{T^p} r^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t) \mid X_t) - r^{\boldsymbol{b}}(\hat{a}_t \mid X_t)\right] \\
&\geq \frac{1}{2^7},
\end{aligned}$$

where in the first inequality we used the fact that π^* always selects the best action in hindsight. Because this holds for any $p \geq 1$, we can use Fatou's lemma to obtain

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t) \mid X_t) - r_t(\hat{a}_t \mid X_t)\right] \geq \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t) \mid X_t) - r_t(\hat{a}_t \mid X_t)\right] \geq \frac{1}{2^7}.$$

As a result, $f.$ is not consistent on the stationary rewards $(r)_t$ under $\mathbb{X}$, which ends the proof of the theorem. ■

A tighter necessary condition $\mathcal{C}_6$ for oblivious rewards

This section proves that $\mathcal{C}_6$ is necessary for stochastic processes, which is tighter than the family $\mathcal{C}_4$. We first prove the lemma on large deviations of the empirical measure in extended CS processes.

Proof of Lemma 6.1 Let $\epsilon > 0$ and suppose by contradiction that for all $T \geq 1$ and $\delta > 0$ there exists a measurable set $A(\delta; T)$ such that $\mathbb{E}[\hat{\mu}_{\mathbb{X}^\mathcal{T}}(A(\delta; T))] \leq \delta$ and

$$\mathbb{E}\left[\sup_{T'>T} \frac{1}{T'} \sum_{t\leq T', t\in\mathcal{T}} \mathbb{1}_{A(\delta;T)}(X_t)\right] > \epsilon.$$

We now construct by induction a sequence of sets $(A_i)_{i\geq 1}$ together with times $(T_i)_{i\geq 0}$ such that $T_0 = 0$. Now suppose that we have constructed T_{i-1} for $i \geq 1$. We take $A_i = A(\epsilon 2^{-i-2}; T_{i-1})$. Then, because $\mathbb{E}[\hat{\mu}_{\mathbb{X}^\mathcal{T}}(A_i)] \leq \epsilon 2^{-i-2}$, by the dominated convergence theorem, there exists $T_i > T_{i-1}$ such that

$$\mathbb{E}\left[\sup_{T>T_i} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_i}(X_t)\right] \leq \frac{\epsilon}{2^{i+1}}.$$

This ends the construction of the sequences. For any $i \geq 1$, let $B_i = A_i \setminus \bigcup_{j<i} A_j$ and note that

$$\begin{aligned}
&\mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_i}(X_t)\right] \\
&\leq \mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{B_i}(X_t)\right] + \sum_{j<i} \mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_j}(X_t)\right] \\
&\leq \mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{B_i}(X_t)\right] + \sum_{j<i} \mathbb{E}\left[\sup_{T>T_j} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_j}(X_t)\right] \\
&\leq \mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{B_i}(X_t)\right] + \frac{\epsilon}{2}.
\end{aligned}$$

By construction $\mathbb{E}\left[\sup_{T>T_{i-1}} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_i}(X_t)\right] > \epsilon$. Hence, letting $C_i = \bigcup_{j\geq i} B_j$, we obtain that for any $j \geq i$,

$$\mathbb{E}\left[\sup_{T>T_j} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{C_i}(X_t)\right] \geq \mathbb{E}\left[\sup_{T>T_j} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{B_{j+1}}(X_t)\right] \geq \frac{\epsilon}{2}.$$

As a result, by the dominated convergence theorem we have $\mathbb{E}[\hat{\mu}_{\mathbb{X}^\mathcal{T}}(C_i)] \geq \frac{\epsilon}{2}$. Further, all sets B_i are disjoint. But $C_i \downarrow \emptyset$, which contradicts the hypothesis that $\mathbb{X}^\mathcal{T} \in \text{CS}$. This ends the proof of the lemma. ■

We recall the necessary definitions to introduce condition $\mathcal{C}_6$. For a process $\mathbb{X} \in \mathcal{C}_4$, any $\epsilon > 0$ and $T \geq 1$,

$$\delta^p(\epsilon; T) := \sup\Bigg\{0 \leq \delta \leq 1 : \forall A \in \mathcal{B} \text{ s.t. } \sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A)] \leq \delta, \\ \forall \tau \geq T \text{ online stopping time}, \quad \mathbb{E}\left[\frac{1}{2\tau}\sum_{\tau\leq t<2\tau, t\in\mathcal{T}^p} \mathbb{1}_A(X_t)\right] \leq \epsilon\Bigg\},$$

and $\delta^p(\epsilon) := \lim_{T\to\infty} \delta^p(\epsilon; T) > 0$. We recall condition $\mathcal{C}_6$.

Condition 6. *$\mathbb{X} \in \mathcal{C}_4$ and for any $\epsilon > 0$, we have $\lim_{p\to\infty} \delta^p(\epsilon) > 0$. Denote by $\mathcal{C}_6$ the set of all processes $\mathbb{X}$ satisfying this condition.*

The main result of this section is that this condition is necessary for oblivious rewards.

Theorem 6.8. *Let $\mathcal{X}$ be a metrizable separable Borel space, and a finite action space $\mathcal{A}$ with $|\mathcal{A}| \geq 2$. Then, $\text{SOAB}_{oblivious} \subset \mathcal{C}_6$.*

Proof Fix $\mathbb{X} \in \mathcal{C}_4 \setminus \mathcal{C}_6$. By hypothesis, there exists $\epsilon > 0$ such that $\delta^p(\epsilon) \to 0$ as $p \to \infty$. Let $(p(i))_{i\geq 1}$ be the set of increasing indices such that $\delta^{p(i)}(\epsilon) \leq \epsilon 2^{-i-3}$. Similarly to the proof of Theorem 6.6, we suppose by contradiction that there is a universally consistent learning rule $f.$ under $\mathbb{X}$ and we will construct by induction some rewards on which the learning rule is not consistent. We will denote by $\hat{a}_t$ the action selected by the learning rule at time t. Precisely, suppose that we have performed $i-1$ iterations of the construction process for some $i \geq 1$, and have constructed times $T^1, \dots, T^{i-1}$ as well as rewards $(r_t)_{t\leq T^{i-1}}$, disjoint sets $A^1, \dots, A^{i-1}$ satisfying

$$\sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A^j)] \leq \epsilon 2^{-j-2}$$

for all $j < i$, and a policy π^* on $\bigcup_{j<i} A^i$. We will now focus on the times $\mathcal{T}^{p(i)}$. For convenience, in the rest of the proof, when clear from context, we will write p instead of $p(i)$.

First, by hypothesis, for any $1 \leq j < i$, we have $\mathbb{E}[\hat{\mu}_{\mathbb{X}^p}(A^j)] \leq \epsilon 2^{-j-2}$. Thus, by the dominated convergence theorem, there exists $t(j)$ such that

$$\mathbb{E}\left[\sup_{T\geq t(j)} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{A^j}(X_t)\right] \leq \frac{\epsilon}{2^{j+1}}.$$

Therefore, summing these equations yields

$$\mathbb{E}\left[\sup_{T\geq \max_{j<i} t(j)} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{\bigcup_{j<i} A^j}(X_t)\right] \leq \frac{\epsilon}{2}.$$

We define $\tilde{T}^{i-1} = \max(T^{i-1}, t(1), \dots, t(i-1))$. Now by construction, $\delta^{p(i)}(\epsilon) \leq \epsilon 2^{-i-3}$. Therefore, there exists $T_0 \geq \tilde{T}^{i-1}$ such that for any $T \geq T_0$, we have $\delta^p(\epsilon; T) \leq \epsilon 2^{-i-2}$. Now for $T \geq T_0$, let $A^i(T) \in \mathcal{B}$ and $\tau^i(T) \geq T$ be a stopping time such that

$$\sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A^i(T))] \leq \epsilon 2^{-i-2} \quad \text{and} \quad \mathbb{E}\left[\frac{1}{2\tau^i(T)}\sum_{\tau^i(T)\leq t<2\tau^i(T), t\in\mathcal{T}^p} \mathbb{1}_{A^i(T)}(X_t)\right] > \epsilon.$$

Last, let $U(T)$ be such that

$$\mathbb{P}[2\tau^i(T) > U(T)] \geq \frac{\epsilon}{2^{T+10}}.$$

Then, by the union bound, with probability at least $1-\epsilon 2^{-10}$, for all $T \geq T_0$, we have $2\tau^i(T) \leq U(T)$. Denote by $\mathcal{H}$ this event. Next, let $k_i = 2^p + 1$, $\alpha_i = 2^{-p-1}$, $\beta_i = \frac{\epsilon}{2^{10}(1+2\alpha_i)^{(k_i-1)k_i 4^{k_i}}}$, $\tilde{K}_i = \left\lceil \frac{2}{\alpha_i} \log \frac{8}{\beta_i} \right\rceil$ and $M_i = \max((1+2\alpha_i)^{\tilde{K}_i}, \frac{2^{10}}{\epsilon})$. We first construct by induction of increasing times $(T(l))_{l\geq 0}$ with $T(0) = M_i T_0$ and $T(l) \geq M_i U(T(l-1))$. For convenience, we use the notation $\tau_l^i = \tau^i(T(l))$, $A_l^i = A^i(T(l)) \setminus \bigcup_{1\leq j<i} A^j$ for $l \geq 0$. Then, by construction, $\tau^i(T(l)) \geq M_i U(T(l-1))$ and

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{2\tau_l^i}\sum_{\tau_l^i\leq t<2\tau_l^i, t\in\mathcal{T}^p} \mathbb{1}_{A_l^i}(X_t)\right] \\
&\geq \mathbb{E}\left[\frac{1}{2\tau_l^i}\sum_{\tau_l^i\leq t<2\tau_l^i, t\in\mathcal{T}^p} \mathbb{1}_{A^i(T(l))}(X_t)\right] - \mathbb{E}\left[\frac{1}{2\tau_l^i}\sum_{\tau_l^i\leq t<2\tau_l^i, t\in\mathcal{T}^p} \mathbb{1}_{\bigcup_{j<i} A^j}(X_t)\right] \\
&> \epsilon - \mathbb{E}\left[\sup_{T\geq \tilde{T}^{i-1}} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_{\bigcup_{j<i} A^j}(X_t)\right] \\
&> \frac{\epsilon}{2}.
\end{aligned}$$

For any $l \geq 1$, let $\delta_l > 0$ such that

$$\mathbb{P}\left[\min_{1\leq t,t'\leq U(T(l)), X_t\neq X_{t'}} \rho(X_t, X_{t'}) \leq \delta_l\right] \leq \frac{\epsilon}{2^{l+10}}.$$

Let $\mathcal{E}$ be the event when for all $l \geq 1$, we have $\min_{1\leq t,t'\leq U(T(l)), X_t\neq X_{t'}} \rho(X_t, X_{t'}) > \delta_l$ and $\mathcal{H}$ is satisfied. By the union bound, $\mathbb{P}[\mathcal{E}] \geq 1 - \frac{\epsilon}{2^9}$. We now construct similar rewards to those in the proof of Theorem 6.6. Then, for any $\delta > 0$ and $u \geq 1$, define the sets $P_u(\delta) = B(x^u, \delta) \setminus \bigcup_{v<u} B(x^v, \delta)$ where $(x^u)_{u\geq 1}$ is a dense sequence of $\mathcal{X}$, which form a

partition of $\mathcal{X}$. For any binary sequence $\boldsymbol{b} = (b_u)_{u\geq 1}$ in $\{0,1\}$ define the deterministic rewards

$$r_{\delta,\boldsymbol{b};l}(a \mid x) = \begin{cases} b_u \mathbb{1}_{x\in A_l^i} & a = a_1, x \in P_u(\delta), \\ \frac{3}{4}\mathbb{1}_{x\in A_l^i} & a = a_2, \\ 0 & a \notin \{a_1, a_2\}. \end{cases}$$

Next, for any sequence of binary sequences $\mathbf{b} := (\boldsymbol{b}^l)_{l\geq 1}$, we construct the deterministic rewards $\boldsymbol{r}^{\mathbf{b}}$ as follows. First, for $t \leq T^{i-1}$, $r_t^{\mathbf{b}} = r_t$ the rewards already constructed. Also, for $T^{i-1} < t \leq U(T(0))$, we pose $r_t^{\mathbf{b}} = 0$. Next, observe that τ_l^i is an online stopping time. Therefore, for any $l \geq 0$, $U(T(l-1)) < t < \tau_l^i$ or $2\tau_l^i \leq t \leq U(T(l))$, we pose $r_t^{\mathbf{b}} = 0$. Finally, for $\tau_l^i \leq t < 2\tau_l^i, U(T(l))$ and k such that $T_p^{k-1} < t \leq T_p^k$, we pose

$$r_t^{\mathbf{b}}(a \mid x_{\leq t}) = \begin{cases} 0 & \exists t' \leq U(T(l-1)) : x_{t'} = x_t, \\ 0 & \text{o.w., } \exists T_p^{k-1} < t' \leq t : x_{t'} = x_t, \\ r_{\delta_l,\boldsymbol{b}^l;l}(a \mid x_t) & \text{o.w., } \forall T_p^{k-1} < t' \leq t : x_{t'} \neq x_t, \end{cases}$$

for any $a \in \mathcal{A}$ and $x_{\leq t} \in \mathcal{X}^t$. Now generate $\mathbf{b}$ as independent i.i.d. Bernoulli $\mathcal{B}(\frac{1}{2})$ processes. We now compare the predictions of the learning rule to the constant policy which selects action a_2. Because the learning rule is consistent under any rewards $\boldsymbol{r}^{\mathbf{b}}$ for any realization $\mathbf{b}$, and because $\mathbb{P}[\mathcal{E}] > 0$, taking the expectation over $\mathbf{b}$, we obtain

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \leq 0.$$

Next, we use the dominated convergence theorem to find $l^i \geq 1$ such that

$$\mathbb{E}\left[\sup_{T\geq T(l_i)/2} \frac{1}{T}\sum_{t=1}^{T} r_t^{\mathbf{b}}(a_2) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \leq \frac{\beta_p}{4}.$$

We now define $A^i = A_{l^i}^i$, $T^i = U(T(l^i))$ and focus on the period $[\tau_l^i, 2\tau_l^i)$. Let $\hat{k} = \max\{k : \tau_l^i \geq T_p^k\}$. Then, $[\tau_l^i, 2\tau_l^i) \subset [T_p^{\hat{k}}, T_p^{\hat{k}+2^p+1})$ and we construct the following sets

$$\mathcal{S}_q = \{T_p^{\hat{k}+q-1} < t \leq T_p^{\hat{k}+q} : X_t \in A^i\} \cap \mathcal{T}^p, \quad 1 \leq q \leq 2^p + 1 = k_i. \tag{6.4}$$

We also define Exp_q as the exploration steps of arm a_1 during $\mathcal{S}_q$.

$$Exp_q = \left\{ t \in \mathcal{S}_q : \hat{a}_t = a_1 \text{ and } \forall t' \in \bigcup_{q'<q} \mathcal{S}_{q'} : X_{t'} = X_t,\ \hat{a}_{t'} \neq a_1 \right\} \\ \setminus \{t : \exists t' \leq U(T(l^i - 1)), X_{t'} = X_t\},$$

and $E_q = |Exp_q|$. The same arguments as in Theorem 6.6 show that for all $1 \leq q \leq k_1$, we have $\mathbb{E}\left[\frac{E_q}{T_p^{\hat{k}+k_i}} \mid \mathcal{E}\right] \leq 4^{q+1}(1+2\alpha_i)^{(k_i-1)k_i}\beta_p$. For any $t \geq 1$, let a_t^* be the optimal action in hindsight and define

$$\mathcal{B}_q = \bigcup_{q\leq\hat{q}} \left\{ t \in \mathcal{S}_q : \forall t' \in \bigcup_{q'<q} \mathcal{S}_{q'} : X_{t'} = X_t, t \notin Exp_{q'} \right\},$$

the times such that we never explored action a_2, before time $T_p^{\hat{k}+q}$. As in the proof of Theorem 6.6, for times in $\mathcal{B}$, the learner incurs an average regret of at least $\frac{1}{8}$. Therefore,

$$\mathbb{E}\left[\frac{1}{T_p^{\hat{k}+k_i}}\sum_{t=1}^{T_p^{\hat{k}+k_i}} r_t^{\mathbf{b}}(a_t^*) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \geq \frac{1}{8}\mathbb{E}\left[\frac{|\mathcal{B}_q|}{T_p^{\hat{k}+k_i}} \mid \mathcal{E}\right].$$

Finally, let $T_p^* = |\{t \leq T_p^{\hat{k}+k_i} : X_t \in A^i\} \cap \mathcal{T}^p|$. Noting that we have $\mathbb{E}\left[\frac{T_p^*}{T_p^{\hat{k}+k_i}} \mid \mathcal{E}\right] \geq \frac{1}{2}\mathbb{E}\left[\frac{T_p^*}{2\tau_l^i} \mid \mathcal{E}\right] \geq \frac{\epsilon}{4} \geq \frac{\epsilon}{16}$, the same arguments as in the original proof give directly

$$\mathbb{E}\left[\frac{1}{T_p^{\hat{k}+k_i}}\sum_{t=1}^{T_p^{\hat{k}+k_i}} r_t^{\mathbf{b}}(a_t^*) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \geq \frac{\epsilon}{2^8}.$$

As a result, there exists a realization of $\mathbf{b}$ such that the above equation holds for this specific realization. We then pose $r_t = r_t^{\mathbf{b}}$ for all $t \leq T^i$ and define a policy π^i on A^i as follows,

$$\pi^i(x) = \begin{cases} a_1 & \text{if } b_u^l = 1, x \in P_u(\delta_{l^i}) \cap A^i, \\ a_2 & \text{if } b_u^l = 0, x \in P_u(\delta_{l^i}) \cap A^i. \end{cases}$$

for any $x \in A^i$, which is possible because A^i is disjoint from $\bigcup_{j<i} A^j$. Now observe that the policy selects the best action in hindsight during the interval $[T(l^i), U(T(l^i))$, irrespective of how it is defined outside of A^i. As a result, we have

$$\begin{aligned}
&\mathbb{E}\left[\sup_{T^{i-1}<T\leq T^i} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t) \mid \mathcal{E}\right] \\
&\geq \mathbb{E}\left[\frac{1}{T^{\hat{k}+k_i}}\sum_{t=1}^{T^{\hat{k}+k_i}} r_t^{\mathbf{b}}(\pi^*(X_t)) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \\
&\geq \mathbb{E}\left[-\frac{2U(T(l^i-1))}{T^{\hat{k}+k_i}} + \frac{1}{T^{\hat{k}+k_i}}\sum_{t=1}^{T^{\hat{k}+k_i}} r_t^{\mathbf{b}}(a_t^*) - r_t^{\mathbf{b}}(\hat{a}_t) \mid \mathcal{E}\right] \\
&\geq -\frac{2}{M_i} + \frac{\epsilon}{2^8} \\
&\geq \frac{\epsilon}{2^9}.
\end{aligned}$$

This ends the recursive construction of the rewards. We close the definition of π^* by setting $\pi^*(x) = a_1$ for $x \notin \bigcup_{i\geq 1} A^i$ arbitrarily. The constructed policy is measurable and we showed that for all $i \geq 1$,

$$\mathbb{E}\left[\sup_{T^{i-1}<T\leq T^i} \frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t(\hat{a}_t)\right] \geq \frac{\epsilon}{2^9}.$$

Using Fatou's lemma, this shows that $\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\tilde{r}_t(\pi^*(X_t))-\tilde{r}_t(\hat{a}_t)\right]\geq\frac{\epsilon}{2^9}$. This ends the proof that $f.$ is not universally consistent under $\mathbb{X}$ and ends the proof of the theorem. ■

We now give an example of process $\mathbb{X}\in\mathcal{C}_4\setminus\mathcal{C}_6$.

Theorem 6.9. *For $\mathcal{X}=[0,1]$ with usual topology, $\mathcal{C}_6\subsetneq\mathcal{C}_4$.*

Proof We construct a process $\mathbb{X}$ on $[0,1]$ by phases $[2^l,2^{l+1})$ for $l\geq 0$. We set $X_1=0$ arbitrarily and divide phases by categories $S_p=\{l\geq 1: l\equiv 2^{p-1}\bmod 2^p\}$ for any $p\geq 1$. Next, for any $l\in S_p$, let

$$A_p(l)=\bigcup_{0\leq i<2^l}\left[\frac{i2^p}{2^{p+l}},\frac{i2^p+1}{2^{p+l}}\right].$$

Importantly, $A_p(l)$ has Lebesgue measure 2^{-p}. Next, noting that $l\geq 2^{p-1}\geq p$, for $2^l\leq t<2^{l+1}$ we define

$$X_t=\begin{cases}\mathcal{U}_t(A_p(l)) & 2^l\leq t<2^l+2^{l-p},\\ X_{t'} & t\geq 2^l+2^{l-p}, 2^l\leq t'<2^l+2^{l-p}, t'\equiv t\bmod 2^{l-p}\end{cases}$$

where $\mathcal{U}_t(A_p(l))$ denotes a uniform random variable on $A_p(l)$ independent from all past random variables. The process on S_p is constructed so that it has 2^p duplicates. This ends the construction of $\mathbb{X}$.

We now show that $\mathbb{X}\in\mathcal{C}_4$. For convenience, for any $l\geq 1$, let $p(l)$ be the index such that $l\in S_{p(l)}$. Next, let $\mathbb{X}^p:=(X_t)_{t\in\mathcal{T}^p}$ for $p\geq 0$. we will show the stronger statement that for any measurable set $A\in\mathcal{B}$, we have $\hat{\mu}_{\mathbb{X}^p}(A)\leq\mu(A)$ $(a.s.)$, where μ is the Lebesgue measure. To do so, fix $A\in\mathcal{B}$ and $\epsilon>0$. Since A is Lebesgue measurable, there exists a sequence of disjoint intervals $(I_k)_{k\geq 0}$ within $\mathcal{X}=[0,1]$ such that $A\subset\bigcup_{k\geq 0}I_k$ and

$$\sum_{k\geq 0}\ell(I_k)\leq\mu(A)+\epsilon,$$

where $\ell(I)$ is the length of an interval I. Then, let k_0 such that $\sum_{k\geq k_0}I_k\leq\frac{\epsilon^2}{2^{p+1}}$ and pose $\ell_0=\min_{k<k_0}\ell(I_k)$. Then, for any $l\geq\max(2,\log_2\frac{k_0}{\epsilon}):=l_0$, with $l\in S_q$,

$$\begin{aligned}\frac{\mu(A\cap A_q(l))}{\mu(A_q(l))}&\leq\sum_{k<k_0}\frac{\mu(I_k\cap A_q(l))}{\mu(A_q(l))}+2^q\mu\left(\bigcup_{k\geq k_0}I_k\right)\\&\leq\sum_{k<k_0}(\ell(I_k)+2^{-l})+\epsilon^2 2^{q-p-1}\\&\leq\mu(A)+2\epsilon+\epsilon^2 2^{q-p-1}.\end{aligned}$$

Let $q_0=p+\log_2\frac{1}{\epsilon}$. For any $l\geq l_0$ with $l\in\bigcup_{q<q_0}S_q$, we have $\frac{\mu(A\cap A_q(l))}{\mu(A_q(l))}\leq\mu(A)+3\epsilon$. Now for any $l\geq l_0$, if $l\in\bigcup_{q<q_0}S_q$, Hoeffding's inequality implies that for any $l\leq r\leq 2^{l-q}$,

$$\mathbb{P}\left[\sum_{2^l\leq t<2^l+r}\mathbb{1}_A(X_t)\leq r(\mu(A)+4\epsilon)\right]\geq 1-e^{-2\epsilon^2r^2}\geq 1-e^{-2\epsilon^2 lr}.$$

Note that we always have $2^{l-q} \geq l$ since $l \geq 2^{q-1}$ and $l \geq 2$. In particular, because we have $\sum_{r\geq 1}\sum_{l\geq 1} e^{-2\epsilon^2 lr} < \infty$, on an event $\mathcal{E}(\epsilon)$ of probability one, there exists $\hat{l} \geq l_0$ such that the above equation holds for all $l \geq \hat{l}$ with $l \in \bigcup_{q<q_0} S_q$ and $l \leq r \leq 2^{l-q}$. Then, for $T \geq 2^{\hat{l}}$, letting $l(T) \geq 1$ such that $2^{l(T)} \leq T < 2^{l(T)+1}$, we have

$$\begin{aligned}
\sum_{t\leq T, t\in\mathcal{T}^p} \mathbb{1}_A(X_t) &= \sum_{l<l(T)} \min(2^{p(l)}, 2^p) \sum_{2^l\leq t<2^l+2^{l-p(l)}} \mathbb{1}_A(X_t) + \sum_{2^{l(T)}\leq t\leq T, t\in\mathcal{T}^p} \mathbb{1}_A(X_t) \\
&\leq \sum_{l<l(T)} \epsilon 2^l \mathbb{1}[p(l) \geq q_0] + 2^{\hat{l}} + \sum_{\hat{l}\leq l<l(T)} 2^l(\mu(A)+4\epsilon)\mathbb{1}[p(l) < q_0] \\
&\quad \epsilon 2^{l(T)}\mathbb{1}[p(l(T)) \geq q_0] + [(T-2^{l(T)}+1)(\mu(A)+4\epsilon) + l(T)]\mathbb{1}[p(l(T)) < q_0] \\
&\leq 2^{\hat{l}} + l(T) + 2\epsilon 2^{l(T)} + (\mu(A)+4\epsilon)T \\
&\leq 2^{\hat{l}} + \log_2 T + (\mu(A)+6\epsilon)T.
\end{aligned}$$

where in the first inequality, we used the fact that for $q \geq q_0$, $2^p \leq \epsilon 2^q$. Further, the additional term $l(T)$ comes from the fact that the estimates on $\mathcal{E}(\epsilon)$ held for $r \geq l$: writing $T = 2^{l(T)} + u2^{l(T)-p(l(T))} + v$, we first use $\mathcal{E}(\epsilon)$ with $r = 2^{l(T)-p(l(T))}$, then with $r = \max(v, l(T))$. As a result, on $\mathcal{E}(\epsilon)$, we have $\hat{\mu}_{\mathbb{X}^p}(A) \leq \mu(A)+6\epsilon$. Thus, on $\bigcap_{j\geq 0}\mathcal{E}(2^{-j})$ of probability one, we have $\hat{\mu}_{\mathbb{X}^p}(A) \leq \mu(A)$, and this holds for all $p \geq 1$ and $A \in \mathcal{B}$. Using this property, verifying the $\mathcal{C}_4$ condition is straightforward. For disjoint measurable sets A_i, we have $\mathbb{E}[\hat{\mu}_{\mathbb{X}^i}(A_i)] \leq \mu(A_i) \to 0$ because $\sum_i \mu(A_i) \leq 1$.

We now show that $\mathbb{X} \notin \mathcal{C}_6$. First, on an event $\mathcal{F}$ of probability one, all samples $\mathcal{U}_t(A_p(l))$ are distinct. As a result, on $\mathcal{F}$, except for the intended duplicates, all instances of $\mathbb{X}$ are distinct. Thus, for any $l \in S_p$, and any $2^l \leq t < 2^{l+1}$, we have $t \in \mathcal{T}^p$. Hence, on $\mathcal{F}$,

$$\frac{1}{2^{l+1}} \sum_{2^l\leq t<2^{l+1}, t\in\mathcal{T}^p} \mathbb{1}_{A_p(l)}(X_t) \geq \frac{2^l}{2^{l+1}} = \frac{1}{2}.$$

In particular, this implies that

$$\mathbb{E}\left[\frac{1}{2^{l+1}} \sum_{2^l\leq t<2^{l+1}, t\in\mathcal{T}^p} \mathbb{1}_{A_p(l)}(X_t)\right] \geq \frac{1}{2}.$$

However, $\mathbb{E}[\hat{\mu}_{\mathbb{X}^p}(A_p(l))] = \mu(A_p(l)) = 2^{-p}$. Therefore, using the trivial stopping time $\tau = 2^l$, we showed $\delta^p(1/2; 2^l) \leq 2^{-p}$. Because this holds for all $l \in S_p$ which is infinite, we have $\delta^p(1/2) \leq 2^{-p}$. Thus, $\delta^p(1/2) \to 0$ as $p \to \infty$. This shows that $\mathbb{X} \notin \mathcal{C}_6$, which ends the proof of the theorem. ■

Further tightened necessary condition $\mathcal{C}_7$ for prescient rewards

A more natural condition on processes than $\mathcal{C}_6$ would be one that does not involve these stopping times τ. In particular, for a process $\mathbb{X} \in \mathcal{C}_4$, we can define instead for any $\epsilon > 0$

and $T \geq 1$,

$$\bar{\delta}^p(\epsilon; T) := \sup \Bigg\{ 0 \leq \delta \leq 1 : \forall A \in \mathcal{B} \text{ s.t. } \sup_l \mathbb{E}[\hat{\mu}_{\mathbb{X}^l}(A)] \leq \delta, \\ \mathbb{E}\left[\sup_{T' \geq T} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^p} \mathbb{1}_A(X_t) \right] \leq \epsilon \Bigg\}.$$

As before, $\bar{\delta}^p(\epsilon; T)$ is non-decreasing in T and $\bar{\delta}^p(\epsilon) := \lim_{T \to \infty} \delta^p(\epsilon; T) > 0$. We can then observe that $\bar{\delta}^p(\epsilon)$ is non-increasing. Similarly to $\mathcal{C}_6$, we can then define the following condition.

Condition 7. *$\mathbb{X} \in \mathcal{C}_4$ and for any $\epsilon > 0$, we have $\lim_{p \to \infty} \bar{\delta}^p(\epsilon) > 0$. Denote by $\mathcal{C}_7$ the set of all processes $\mathbb{X}$ satisfying this condition.*

As a simple remark, we have the inclusion $\mathcal{C}_7 \subset \mathcal{C}_6$, since if for any given process $\mathbb{X} \in \mathcal{C}_4$, set $A \in \mathcal{B}$ and online stopping time $\tau \geq T$,

$$\mathbb{E}\left[\frac{1}{2\tau} \sum_{\tau \leq t < 2\tau, t \in \mathcal{T}^p} \mathbb{1}_A(X_t) \right] \leq \mathbb{E}\left[\sup_{T' \geq T} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^p} \mathbb{1}_A(X_t) \right].$$

Unfortunately, for oblivious rewards, we were unable to prove that $\mathcal{C}_7$ is a necessary condition. Indeed, for a process $\mathbb{X} \in \mathcal{C}_4$, time $T \geq 1$ and $\epsilon > 0$, if

$$\mathbb{E}\left[\sup_{T' \geq T} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^p} \mathbb{1}_A(X_t) \right] > \epsilon, \tag{6.5}$$

it is in general not true that there exists an online stopping time $\tau \geq T$ such that

$$\mathbb{E}\left[\frac{1}{2\tau} \sum_{\tau \leq t < 2\tau, t \in \mathcal{T}^p} \mathbb{1}_A(X_t) \right] > \eta\epsilon, \tag{6.6}$$

even for a fixed multiplicative tolerance $0 < \eta < 1$, which should be independent of $\epsilon > 0$. Thus, it seems unlikely that $\mathcal{C}_6 = \mathcal{C}_7$ in general for spaces $\mathcal{X}$ admitting a non-atomic probability measure.

However, if one considers a stronger type of adversary, we can show that $\mathcal{C}_7$ becomes necessary for universal learning. Precisely, one can introduce *prescient* rewards, that are stronger than oblivious rewards in that rewards are allowed to depend on the complete sequence $\mathbb{X}$ instead of the revealed contexts to the learner $\mathbb{X}_{\leq t}$ at step t. Formally, these are defined as follows.

Definition 6.5 (Reward models)**.** *The reward mechanism is said to be* prescient *if there are conditional distributions $(P_{r|a,\boldsymbol{x}_{t' \geq 1}})_{t \geq 1}$ such that r_t given the selected action a_t and the sequence of contexts $\mathbb{X}$, follows $P_{r|a,\boldsymbol{x}_{t' \geq 1}}$.*

In this model, given a process $\mathbb{X} \in \mathcal{C}_4$, a time $T \geq 1$ and $\epsilon > 0$ satisfying Eq (6.5), finding a time $\tau \geq T$ (measurable with respect to the sigma-algebra $\sigma(\mathbb{X})$, i.e., conditionally on $\mathbb{X}$) such that Eq (6.6) is satisfied becomes trivial even with $\eta = 1$. Therefore, the same proof as for Theorem 6.8 shows that the last condition on stochastic processes is necessary for prescient rewards.

Theorem 6.10. *Let $\mathcal{X}$ be a metrizable separable Borel space, and a finite action space $\mathcal{A}$ with $|\mathcal{A}| \geq 2$. Then, $\textsc{SOAB}_{prescient} \subset \mathcal{C}_7$.*

Condition $\mathcal{C}_5$ is necessary for universal learning with online rewards

In this section, we show that condition $\mathcal{C}_5$ is necessary for universal learning with online rewards, tightening the result on the necessity of condition $\mathcal{C}_6$ from the previous section. In fact, in Section 6.5.2 we show that $\mathcal{C}_5$ is also sufficient, which together with the result from this section shows that $\mathcal{C}_5$ exactly characterizes universally learnable processes for online rewards. We recall that this is the strongest reward model that we consider in this chapter and allows the reward adversary to also take into account the past actions selected by the learner. We first briefly recall the definition of condition $\mathcal{C}_5$.

Condition 5. *There exists an increasing sequence of integers $(T_i)_{i\geq 0}$ such that letting*

$$\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\},$$

we have $\mathbb{X}^{\mathcal{T}} \in CS$. Denote by $\mathcal{C}_5$ the set of all processes $\mathbb{X}$ satisfying this condition.

Before proving our main result, we need the following lemma that gives an equivalent formulation of the class of processes $\mathcal{C}_5$. Intuitively, it shows that if $\mathbb{X} \notin \mathcal{C}_5$, for any tentative rate to add duplicates—yielding the extended process $\tilde{\mathbb{X}}$—we can uniformly lower-bound the proportion of failure for the CS condition.

Lemma 6.3. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathbb{X}$ a stochastic process on $\mathcal{X}$. The following are equivalent.*

- $\mathbb{X} \in \mathcal{C}_5$,
- *For any $\epsilon > 0$, there exists an increasing sequence of integers $(T_i)_{i\geq 0}$ such that letting $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$, for any sequence $\{A_k\}_{k\geq 1}$ of measurable sets of $\mathcal{X}$ with $A_k \downarrow \emptyset$,*

$$\lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_k)] \leq \epsilon.$$

Proof By definition of the condition $\mathcal{C}_5$, it is immediate that $\mathbb{X} \in \mathcal{C}_5$ implies the second proposition. It remains to prove the converse. We then suppose that $\mathbb{X}$ satisfies the second proposition. Denote by $(T_i(l))_{i\geq 0}$ the sequence obtained from the proposition by setting $\epsilon = 2^{-l}$. Now defining

$$T_i = \max_{j\leq i} T_i(j),$$

it then suffices to argue that the sequence $(T_i)_{i\geq 0}$ satisfies the requirements for the $\mathcal{C}_5$ condition. We write $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$ and $\mathcal{T}(l) = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i(l)\}$ for any $l \geq 0$. Now fix $l \geq 0$, and note that for any $i \geq l$, one has $T_i \geq T_i(l)$. As a result,

$$\bigcup_{i\geq l} \mathcal{T}^i \cap \{t \geq T_i\} \subset \bigcup_{i\geq l} \mathcal{T}^i \cap \{t \geq T_i(l)\}.$$

Next, note that because the sets $\mathcal{T}^i$ are increasing in i, we have $\mathcal{T} \setminus \bigcup_{i\geq l} \mathcal{T}^i \cap \{t \geq T_i\} \subset \{t < T_l\}$. Therefore, for any measurable set $A \in \mathcal{B}$, one has

$$\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_A(X_t) \leq \limsup_{T\to\infty} \frac{T_l}{T} + \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}(l)} \mathbb{1}_A(X_t) = \hat{\mu}_{(X_t)_{t\in\mathcal{T}(l)}}(A).$$

Thus, for any sequence of measurable sets $A_k \downarrow \emptyset$, one has

$$\lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_k)] \leq \lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}(l)}}(A_k)] \leq 2^{-l}.$$

Because this holds for all $l \geq 0$, we obtain $\lim_{k\to\infty} \mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_k)] = 0$ and the lemma is proved. ■

We are now ready to prove the following theorem.

Theorem 6.11. *Let $\mathcal{X}$ be a metrizable separable Borel space, and a finite action space $\mathcal{A}$ with $|\mathcal{A}| \geq 2$. Then, $\textsc{SOAB}_{online} \subset \mathcal{C}_5$.*

Proof Fix $\mathbb{X} \notin \mathcal{C}_5$. If $\mathbb{X} \notin \mathcal{C}_4$, we already proved that (even for oblivious rewards) universal learning is not achievable. Therefore, we suppose that $\mathbb{X} \in \mathcal{C}_4$ and suppose by contradiction that there is a universally consistent learning rule f. under $\mathbb{X}$. We will construct by induction some online rewards on which the learning rule is not consistent. For convenience, we denote by $\hat{a}_t$ the action selected by the learning rule at time t. Last, since $|\mathcal{A}| \geq 2$, we can fix $a_1 \neq a_2 \in \mathcal{A}$ two arbitrary actions. These will be the only used actions for our constructions, all other actions $a \in \mathcal{A} \setminus \{a_1, a_2\}$ will have zero rewards at all times.

We start by constructing rewards that will depend on the actions of the learning rule. By Lemma 6.3, we can fix ϵ such that for any increasing sequence $(T_i)_{i\geq 0}$, letting $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$, there exists a sequence of sets $A_k \downarrow \emptyset$ such that

$$\mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_k)] \geq \epsilon, \quad \forall k \geq 0.$$

Here we used that the sequence of sets is decreasing so that $\mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_k)]$ is decreasing in i.

The end rewards are constructed by induction: at the phase p of the construction, the rewards $r^\star_t$ have been constructed for all $t < T^\star_p$ for some time $T^\star_p = 2^{R^\star_p}$. Further, we have defined some disjoint sets $B_1, \dots, B_p$, increasing times $T^\star_1, \dots, T^\star_{p-1}$, and a policy $\pi^{(p)}$ such that $\pi^{(p)}(x) = a_2$ for all $x \notin B_1 \cup \cdots B_p$, and for any $p' \leq p$,

$$\mathbb{E}\left[\max_{T^\star_{p'-1}\leq T<T^\star_{p'}} \frac{1}{T} \sum_{t=1}^{T} r^\star_t(\pi^{(p)}(X_t)) - r^\star_t(\hat{a}_t)\right] \geq \frac{\epsilon}{16} + \frac{\epsilon}{2^{p+10}}, \tag{6.7}$$

where we used the notation $T_0^\star = 0$. Last, at phase p we have also constructed a sequence of increasing indices $(Q_p(i))_{i\geq 0}$ with $Q_p(i) \geq 4i$ such that with $\mathcal{T}^{(p)} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq 2^{Q_p(i)}\}$, one has

$$\mathbb{E}\left[\sup_{T\geq 1} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{(p)}} \mathbb{1}_{B_{p'}}(X_t)\right] \leq \frac{\epsilon}{2^{p'+10}}, \quad p' \leq p. \tag{6.8}$$

For instance, for $p = 0$ we can simply take $Q_0(i) = 2i$ for all $i \geq 0$. We then suppose that we completed phase $p \geq 0$ and proceed with the induction to construct the set B_{p+1}, time $T_{p+1}^\star$ and rewards $r_t^\star$ until time $T_{p+1}^\star$.

Before doing so, we need to construct an auxiliary reward process. These rewards have the following behavior. Before $T_p^\star = 2^{R_p^\star}$, these are constructed identically as the rewards $\boldsymbol{r}^\star$. Then, at time $t \geq 2^{R_p^\star}$, either the rewards are always zero and this is called an inactive time; or the time is active, in which case the "safe" action a_2 always receives a reward $3/4$, and the "uncertain" action a_1 receives a reward that can either be 0 or 1 with equal probability. We say that the learning rule explores at an active time t if it selects action a_1. At the high level, the rewards proceed by period and tentatively activate the times from $\mathcal{T}^i$ for some $i \geq 0$. If the learning rule performs too many explorations, the trial fails and we instead aim to activate fewer times from $\mathcal{T}^j$ for $j < i$. We construct the rewards inductively by period $[2^r, 2^{r+1})$ for $r \geq r_0$. Each of these periods will be associated with a level $i(r) \geq 0$, which roughly corresponds to the fact that the active times during period r were times in $\mathcal{T}^{i(r)}$. We also denote by $\mathcal{S}_t$ the set of active times up until time t (included). The formal procedure to define the online rewards is given in Algorithm 6.1, where $r_t(a)$ denotes the reward for action a defined by the procedure at time t, for $t \geq 1$.

Let $\mathcal{S} = \bigcup_{t\geq 1} \mathcal{S}_t$ be the set of all active times. We first give some properties on the learning procedure starting from time $T_p^\star$. As a first step, we show that the learner cannot make better predictions than the simple policy $\pi_0 : x \in \mathcal{X} \mapsto a_2 \in \mathcal{A}$. Precisely, we show that the quantities $r_t(\hat{a}_t) - r_t(a_2) + \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t\neq a_2}/4$ for $t \geq T_p^\star$ form the increments of a super-martingale with respect to the filtration $\sigma(\mathbb{X}_{\leq t}, \hat{\boldsymbol{a}}_{\leq t}, \boldsymbol{r}_{\leq t-1})$. First, note that whether t is active, i.e., $t \in \mathcal{S}$ only requires the knowledge of $\mathbb{X}_{\leq t}$ and the actions $\hat{\boldsymbol{a}}_{\leq t}$, hence is measurable with respect to the given filtration. Next, if t is inactive, all rewards are zero. We now consider active times. Denote by $u(t)$ the time of the first occurrence of X_t starting from $T_p^\star$, i.e., $u(t) = \min\{T_p^\star \leq u \leq t : X_t = X_u\}$. Then, if t is active, $r_t(a_1) - r_t(a_2) = B_{u(t)} - 3/4$. Moreover, by construction, the learning rule has not queried a_1 for any previous active time u within the same period as t such that $X_t = X_u$. However, these are the only times when $B_{t'}$ affected the rewards. As a result, all rewards that the learning rule has received before time t are independent of $B_{u(t)}$ (whether t is active or not). This shows that $B_{u(t)}$ is independent from $\mathbb{X}_{\leq t}$, $\hat{\boldsymbol{a}}_{\leq t}$ and $\boldsymbol{r}_{\leq t-1}$ together. As a result,

$$\begin{aligned}
&\mathbb{E}[r_t(\hat{a}_t) - r_t(a_2) + \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t\neq a_2}/4 \mid \mathbb{X}_{\leq t}, \hat{\boldsymbol{a}}_{\leq t}, \boldsymbol{r}_{\leq t-1}] \\
&\qquad = \mathbb{1}_{t\in\mathcal{S}}(-1/2 \cdot \mathbb{1}_{\hat{a}_t\notin\{a_1,a_2\}} + \mathbb{1}_{\hat{a}_t=a_1}\mathbb{E}[B_{u(t)} - 1/2 \mid \mathbb{X}_{\leq t}, \hat{\boldsymbol{a}}_{\leq t}, \boldsymbol{r}_{\leq t-1}]) \\
&\qquad = -1/2 \cdot \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t\notin\{a_1,a_2\}} \leq 0.
\end{aligned}$$

This ends the proof that $(r_t(\hat{a}_t) - r_t(a_2) + \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t\neq a_2}/4)_{t\geq T_p^\star}$ form the increments of a super-martingale, and these are bounded in absolute value by one. Azuma-Hoeffding's inequality

Let $(B_t)_{t\geq 1}$ be an i.i.d. $\mathcal{B}(\frac{1}{2})$ sequence
for $t = 1, \ldots, T_p^\star - 1$ **do**
  Observe context X_t
  Define $r_t(a) = r_t^\star(a)$ for all $a \in \mathcal{A}$
  Observe action selected by learner $\hat{a}_t$
end
Initialize $i(R_p^\star) = 0$ and let $\mathcal{S}_{T_p^\star - 1} = \emptyset$
for $r \geq R_p^\star$ **do**
  for $t = 2^r, \ldots, 2^{r+1} - 1$ **do**
    Observe context X_t
    if $t \notin \mathcal{T}^{i(r)}$ **then**
      Define $r_t(a) = 0$ for all $a \in \mathcal{A}$ and $\mathcal{S}_t = \mathcal{S}_{t-1}$
    else if $\forall T_p^\star \leq t' < t$, $X_{t'} \neq X_t$ **then**
      Define $r_t(a) = \begin{cases} B_t & a = a_1 \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\} \end{cases}$ for $a \in \mathcal{A}$
      $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{t\}$
    else if $\exists T_p^\star \leq t' < t$ *such that* $X_t = X_{t'}$, $t' \in \mathcal{S}_{t-1}$ *and* $\hat{a}_{t'} = a_1$ **then**
      Define $r_t(a) = 0$ for all $a \in \mathcal{A}$ and $\mathcal{S}_t = \mathcal{S}_{t-1}$
    else
      Define $r_t(a) = r_{t'}(a)$ for all $a \in \mathcal{A}$ where $t' < t$, $X_t = X_{t'}$ and $t' \in \mathcal{S}_{t-1}$
      $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{t\}$
    end
    Observe action selected by learner $\hat{a}_t$
    while $\frac{1}{t}\sum_{u=T_p^\star}^{t} \mathbb{1}_{u\in\mathcal{S}_t}\mathbb{1}_{\hat{a}_u \neq a_2} \geq \frac{1}{2^{2i(r)}(i(r)+1)}$ **do** $i(r) \leftarrow \max(0, i(r) - 1)$;
  end
  Define $i(r+1) = \min\{i(r) + 1, k\}$ where k is such that $Q_p(k) \leq r + 1 < Q_p(k+1)$
end

Algorithm 6.1: Procedure to define the online rewards

then implies for any $T \geq T_p^\star$,

$$\mathbb{P}\left[\sum_{t=T_p^\star}^{T} r_t(\hat{a}_t) - r_t(a_2) \geq 2T^{3/4} - \frac{1}{4}\sum_{t=T_p^\star}^{T} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t \neq a_2}\right] \leq e^{-2\sqrt{T}}.$$

Borel-Cantelli's lemma then implies that on an event $\mathcal{E}$ of probability one, there exists $\hat{T} \geq T_p^\star$ such that for any $T \geq \hat{T}$,

$$\sum_{t=T_p^\star}^{T} r_t(\hat{a}_t) - r_t(a_2) < 2T^{3/4} - \frac{1}{4}\sum_{t=T_p^\star}^{T} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t \neq a_2}.$$

We now focus on the level $i(r)$ at each period. Note that this quantity is updated by the procedure along the learning process: it starts at $i(r-1)+1$ (or 0 if $r = r_0$) at the

beginning of the period $[2^r, 2^{r+1})$, then can only decrease during the period. Starting from the end of the period 2^{r+1}, the level $i(r)$ is never updated again. To avoid any confusion, we denote by $I(r)$ this final value of $i(r)$ once the period is completed. We aim to prove that the level at each period $i(r)$ eventually diverges to infinity. Fix $j \geq 0$. Because $f_\cdot$ is universally consistent under $\mathbb{X}$, it has in particular vanishing excess error compared to π_0. Hence, we have

$$\mathbb{P}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_t(a_2) - r_t(\hat{a}_t) \geq \frac{1}{2^{2j+4}(j+1)}\right] = 0.$$

As a result, by the dominated convergence theorem there exists $t_j \geq 1$ such that

$$\mathbb{P}\left[\sup_{T\geq t_j} \frac{1}{T}\sum_{t=1}^{T} r_t(a_2) - r_t(\hat{a}_t) \geq \frac{1}{2^{2j+4}(j+1)}\right] \leq \frac{\epsilon}{2^{j+10}}.$$

We denote by $\mathcal{F}_j$ the complement event. Next, because $\mathcal{E}$ has full probability, there exists t'_j such that

$$\mathbb{P}\left[\sum_{t=T_p^\star}^{T} r_t(\hat{a}_t) - r_t(a_2) < 2T^{3/4} - \frac{1}{4}\sum_{t=T_p^\star}^{T} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{\hat{a}_t\neq a_2},\ \forall T \geq t'_j\right] \leq \frac{\epsilon}{2^{j+10}}.$$

We denote by $\mathcal{E}_j$ the complement event. Now, we define an integer $R_j \geq R_p^\star$ such that $2^{R_j-j} \geq \max(t_j, t'_j, 2^{8j+16}(j+1)^4, 2^{2j+4}(j+1)T_p^\star, 2^{Q_p(j)})$. Using the previous two equations shows that on $\mathcal{E}_j \cap \mathcal{F}_j$ of probability at most $1 - \frac{\epsilon}{2^{j+9}}$, for all $T \geq 2^{R_j-j}$,

$$\begin{aligned}\frac{1}{T}\sum_{t=T_p^\star}^{T} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{t\neq a_2} &< \frac{4}{T}\sum_{t<T_p^\star}(r_t(\hat{a}_t) - r_t(a_2)) + \frac{8}{T^{1/4}} + \frac{1}{2^{2j+2}(j+1)} \\ &\leq \frac{4T_p^\star}{T} + \frac{8}{T^{1/4}} + \frac{1}{2^{2j+2}(j+1)} \leq \frac{1}{2^{2j}(j+1)}.\end{aligned}$$

Also, for any $r \geq R_j - j$, one has $r \geq Q_p(j)$ so that the quantities $I(r)$ can freely increase until they reach j from when the quantities $i(r)$ are always lower bounded by j. In particular, using the union bound, we obtain

$$\mathbb{P}\left[\forall j \geq 0, \inf_{r\geq R_j} I(r) \geq j\right] \geq \mathbb{P}\left[\bigcap_{j\geq 0} \mathcal{E}_j \cap \mathcal{F}_j\right] \geq 1 - \frac{\epsilon}{2^8}.$$

We denote by $\mathcal{F} = \{\forall j \geq 0, \inf_{r\geq R_j} I(r) \geq j\}$ the corresponding event.

We are now ready to show that $f_\cdot$ is not universally consistent. Because $\mathbb{X} \notin \mathcal{C}_5$, with $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq 2^{R_j}\}$, there exists a measurable sets $A_k \downarrow \emptyset$ such that for all $k \geq 1$ we have $\mathbb{E}[\widehat{\widehat{\mu}}_{(X_t)_{t\in\mathcal{T}}}(A_k)] \geq \epsilon$. Now because $A_k \downarrow \emptyset$, we have

$$0 \leq \lim_{k+\to\infty} \mathbb{P}\left(\exists t < T_p^\star : X_t \in A_k\right) \leq \sum_{t<T_p^\star} \lim_{k\to\infty} \mathbb{P}(X_t \in A_k) = 0.$$

Also, because $\mathbb{X} \in \mathcal{C}_4$, by Proposition 6.5 we have

$$\lim_{k\to\infty} \mathbb{E}\left[\sup_{i\geq 0} \hat{\mu}_{(X_t)_{t\in\mathcal{T}^i}}(A_k)\right] = 0.$$

As a result, there exists $l \geq 1$ such that

$$\mathbb{E}\left[\sup_{i\geq 0} \hat{\mu}_{(X_t)_{t\in\mathcal{T}^i}}(A_l)\right] \leq \frac{\epsilon}{2^{p+11}} \quad \text{and} \quad \mathbb{P}\left(\exists t < T_p^\star : X_t \in A_k\right) \leq \frac{\epsilon}{2^{p+11}}. \tag{6.9}$$

We fix this index l in the rest of the proof. Let $L_p^\star$ be an integer such that $L_p^\star \geq \max(R_p^\star + 10 - \log_2\epsilon, R_{10-\log_2\epsilon}, 4(\log_2(C_\epsilon) + 10 - \log_2\epsilon))$, where $C_\epsilon = \sqrt{2\ln\frac{8}{\epsilon}}$. Now by construction, since we have $\mathbb{E}[\hat{\mu}_{(X_t)_{t\in\mathcal{T}}}(A_l)] \geq \epsilon$, we have in particular

$$\mathbb{E}\left[\sup_{T\geq 2^{L_p^\star}} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t)\right] \geq \epsilon.$$

Thus, by the dominated convergence theorem, there exists an integer $R_{p+1}^\star > 2^{L_p^\star}$ such that

$$\mathbb{E}\left[\max_{2^{L_p^\star}\leq T<2^{R_{p+1}^\star}} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t)\right] \geq \frac{\epsilon}{2}. \tag{6.10}$$

We define $T_{p+1}^\star = 2^{R_{p+1}^\star}$. As a second step, we show that during the learning process until time $T_{p+1}^\star$, for a large proportion of active times t for which $X_t \in A_l$, the optimal arm in hindsight is a_1. Precisely, we aim to show that

$$\mathbb{E}\left[\max_{2^{L_p^\star}\leq T<T_{p+1}^\star} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{S}} \mathbb{1}_{A_l}(X_t)\cdot B_{u(t)}\right] \geq \frac{\epsilon}{8}.$$

To prove this, we reason conditionally on $\mathbb{X}$. Define

$$\hat{T} = \operatorname*{arg\,max}_{2^{L_p^\star}\leq T<T_{p+1}^\star} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t).$$

Also, let $Exp = \{T_p^\star \leq t \leq \hat{T} : t \in \mathcal{S}, X_t \in A_l, \hat{a}_t = a_1\}$ the set of "exploration" times on A_l when the learning rule selected action a_1 without prior knowledge on the value $B_{u(t)}$ for active time t. For any exploration time $t \in Exp$, we also define $N(t) = |\{T_p^\star \leq t' \leq t : t \in \mathcal{S}, X_{t'} = X_t\}|$ the number of active occurrences of X_t before the exploration at t. Note that after the exploration, new duplicates of X_t will never be active anymore. Last, denote by $Unexp = (A_l \cap \{X_t, T_p^\star \leq t \leq \hat{T}\}) \setminus \{X_t, t \in Exp\}$ the set of points in A_l that were left unexplored until horizon $\hat{T}$. As above, for $x \in Unexp$, we denote by $N(x) = |\{T_p^\star \leq t \leq \hat{T} : t \in \mathcal{S}, X_{t'} = x|$ the number of active occurrences of x until $\hat{T}$. Also, by abuse of notation, for any $x \in Unexp$, we denote $u(x) = \min\{T_p^\star \leq t \leq \hat{T} : X_t = x\}$ the first occurrence of X_t. Conditionally on the realization of $\mathbb{X}$ (which as a result makes $\hat{T}$ deterministic), the

sequence $(\mathbb{1}_{t\in Exp}N(t)(B_{u(t)}-\frac{1}{2}))_{T_p^\star\le t\le\hat{T}}$ followed by the sequence $(N(x)(B_{u(x)}-\frac{1}{2}))_{x\in Unexp}$ form the increments of a martingale with filtration given by the σ-algebras $\sigma(\mathbb{X},\hat{\boldsymbol{a}}_{\le t},\boldsymbol{r}_{\le t-1})$. Indeed, conditionally on $\mathbb{X}$, the past history $\hat{\boldsymbol{a}}_{\le t-1},\boldsymbol{r}_{\le t-1}$ and the selected action $\hat{a}_t$, at an exploration time $t\in Exp$, the value $B_{u(t)}$ is independent of $\mathbb{X}$ and has never been revealed yet, hence is independent of the history as well. Similarly, for unrevealed points $x\in Unexp$, the variables $B_{u(x)}$ are together independent and also independent from $\mathbb{X}$ and the history $\hat{\boldsymbol{a}}_{\le\hat{T}},\boldsymbol{r}_{\le\hat{T}}$. The final term of the described martingale writes

$$\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in Exp}N(t)\left(B_{u(t)}-\frac{1}{2}\right)+\sum_{x\in Unexp}N(x)\left(B_{u(x)}-\frac{1}{2}\right)$$
$$=\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)\left(B_{u(t)}-\frac{1}{2}\right).$$

We now bound these increments. For any $R_p^\star\le r<R_{p+1}^\star$, during the period $[2^r,2^{r+1})$, one has $\mathcal{S}\cap[2^r,2^{r+1})\subset\mathcal{T}^k$, where k is such that $Q_p(k)\le r<Q_p(k+1)$. Now recall that $Q_p(k)\ge 4k$ so that the number of active duplicates for a given point x during period r is at most $2^k\le 2^{r/4}$. Hence, if $\hat{T}\in[2^{\hat{r}},2^{\hat{r}+1})$, the number of active duplicates of any point until $\hat{T}$ satisfies

$$\max_{t\in Exp}N(t),\max_{x\in Unexp}N(x)\le\sum_{r=r_0}^{\hat{r}}2^{r/4}\le\frac{2^{r/4}}{1-2^{-1/4}}\le\frac{\hat{T}^{1/4}}{2^{1/4}-1}\le 6\hat{T}^{1/4}.$$

In particular, all increments of the constructed martingale have elements norm bounded by the above value. Azuma-Hoeffding's inequality then yields

$$\mathbb{P}\left[\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)\left(B_{u(t)}-\frac{1}{2}\right)\le -C_\epsilon\hat{T}^{3/4}\mid\mathbb{X}\right]\le\frac{\epsilon}{8}.$$

Let $\mathcal{G}$ be the complement event, i.e., the event when $\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)\left(B_{u(t)}-\frac{1}{2}\right)>-C_\epsilon\hat{T}^{3/4}$. Then, using Eq (6.10) we obtain

$$\mathbb{E}\left[\frac{\mathbb{1}_{\mathcal{F}\cap\mathcal{G}}}{\hat{T}}\sum_{t\le\hat{T},t\in\mathcal{T}}\mathbb{1}_{A_l}(X_t)\right]\ge\mathbb{E}\left[\frac{1}{\hat{T}}\sum_{t\le\hat{T},t\in\mathcal{T}}\mathbb{1}_{A_l}(X_t)\right]-\mathbb{P}[\mathcal{F}]-\mathbb{P}[\mathcal{G}]\ge\frac{\epsilon}{2}-\frac{\epsilon}{8}-\frac{\epsilon}{8}=\frac{\epsilon}{4}.\quad(6.11)$$

As a last step, we show that under $\mathcal{F}\cap\mathcal{G}$, the learning rule incurs significant regret compared to the best action in hindsight for times with contexts falling in A_l. On $\mathcal{F}\cap\mathcal{G}$,

$$\frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)B_{u(t)}\ge\frac{1}{2\hat{T}}\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)-\frac{C_\epsilon}{\hat{T}^{1/4}}\ge\frac{1}{2\hat{T}}\sum_{t=T_p^\star}^{\hat{T}}\mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)-\frac{\epsilon}{2^{10}}.$$

We used $\hat{T}\ge 2^{L_p^\star}$ in the last inequality. We now aim to compare the right-hand side of the last inequality to $\frac{1}{\hat{T}}\sum_{T_p^\star\le t\le\hat{T},t\in\mathcal{T}}\mathbb{1}_{A_l}(X_t)$. Because $\mathcal{F}$ is satisfied, $\mathcal{T}\setminus\mathcal{S}$ the set of inactive

times that are counted within $\mathcal{T}$ only contains times t such that there exists $t' < t$ with $t' \in \mathcal{S}$ when the learning rule performed an exploration (see Algorithm 6.1). Thus,

$$\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}} \mathbb{1}_{A_l}(X_t) \geq \sum_{t\leq\hat{T}, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t) - T_p^\star - \sum_{t\in Exp} |\{t < t' \leq \hat{T}, t' \in \mathcal{T}\setminus\mathcal{S}, X_{t'} = X_t\}|.$$

Letting $\hat{j}$ be the integer such that $R_{\hat{j}} \leq \hat{R} < R_{\hat{j}+1}$, i.e., $2^{R_{\hat{j}}} \leq \hat{T} < 2^{R_{\hat{j}+1}}$, we observe that

$$\begin{aligned}
&\sum_{t\in Exp} |\{t < t' \leq \hat{T}, t' \in \mathcal{T}\setminus\mathcal{S}, X_{t'} = X_t\}| \\
&\qquad \leq 2^{\hat{R}-\hat{j}} + \sum_{t\in Exp} |\{2^{\hat{R}-\hat{j}}, t < t' \leq \hat{T}, t' \in \mathcal{T}^{\hat{j}}, X_{t'} = X_t\}| \\
&\qquad \leq 2^{\hat{R}-\hat{j}} + |Exp| 2^{\hat{j}}(\hat{j}+1) \\
&\qquad \leq \frac{\hat{T}}{2^{\hat{j}-1}} + |Exp| 2^{\hat{j}}(\hat{j}+1).
\end{aligned}$$

where we used the fact that because $(R_j)_{j\geq 1}$ is increasing, each distinct point is duplicated at most $2^{\hat{j}}$ times in any period $\mathcal{T}\cap[2^r, 2^{r+1})$ with $r < R_{\hat{j}+1}$. Next, because $\mathcal{F}$ is satisfied we have in particular $I(\hat{R}) \geq \hat{j}$, implying that at time $\hat{T}$, we had the guarantee

$$\frac{|Exp|}{\hat{T}} \leq \frac{1}{\hat{T}} \sum_{u=T_p^\star}^{\hat{T}} \mathbb{1}_{u\in\mathcal{S}} \mathbb{1}_{\hat{a}_u \neq a_2} < \frac{1}{2^{2I(\hat{R})}(I(\hat{R})+1)} \leq \frac{1}{2^{2\hat{j}}(\hat{j}+1)}.$$

Combining the previous four equations and the fact that $\hat{T} \geq 2^{L_p^\star}$ shows that on $\mathcal{F}\cap\mathcal{G}$ one has

$$\begin{aligned}
\frac{1}{\hat{T}} \sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}} \mathbb{1}_{A_l}(X_t) B_{u(t)} &\geq \frac{1}{2\hat{T}} \sum_{t\leq\hat{T}, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t) - \frac{\epsilon}{2^{10}} - \frac{T_p^\star}{2\hat{T}} - \frac{1}{2^{\hat{j}}} - \frac{1}{2^{\hat{j}+1}} \\
&\geq \frac{1}{2\hat{T}} \sum_{t\leq\hat{T}, t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t) - \frac{\epsilon}{2^8}.
\end{aligned}$$

In the last inequality, we used $\hat{j} \geq 10 - \log_2 \epsilon$, a consequence of $\hat{T} \geq 2^{L_p^\star}$. We are now ready to compare the reward of the learning rule to the best action in hindsight for times t such that $X_t \in A_l$. Precisely, consider the following actions $a_t^\star$: at an active time $t \in \mathcal{S}$ and $X_t \in A_l$, we pose $a_t^\star = a_1$ if $B_{u(t)} = 1$ and $a_t^\star = a_2$ otherwise. For any other active time $t \in \mathcal{S}$ and $X_t \notin A_l$, we pose $a_t^\star = a_2$ (which is in that case not necessarily the best action in

hindsight). First note that

$$\begin{aligned}\frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{A_l}(X_t)(r_t(a_t^\star) - r_t(\hat{a}_t)) &= \frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)\left(\frac{3+B_{u(t)}}{4} - r_t(\hat{a}_t)\right)\\ &\geq \frac{1}{4\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)\mathbb{1}_{t\notin Exp}B_{u(t)}\\ &\geq \frac{1}{4\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)B_{u(t)} - \frac{1}{4\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in Exp}\mathbb{1}_{A_l}(X_t).\end{aligned}$$

Also, note that

$$\begin{aligned}\frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{A_l^c}(X_t)(r_t(a_t^\star) - r_t(\hat{a}_t)) &\geq \frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l^c}(X_t)\left(\frac{3}{4} - r_t(\hat{a}_t)\right)\\ &\geq -\frac{1}{4\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in Exp}\mathbb{1}_{A_l^c}(X_t).\end{aligned}$$

Combining the two previous equations shows that on $\mathcal{F}\cap\mathcal{G}$,

$$\frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} r_t(a_t^\star) - r_t(\hat{a}_t) \geq \frac{1}{4\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} \mathbb{1}_{t\in\mathcal{S}}\mathbb{1}_{A_l}(X_t)B_{u(t)} - \frac{|Exp|}{4\hat{T}} \geq \frac{1}{2\hat{T}}\sum_{t\leq\hat{T},t\in\mathcal{T}} \mathbb{1}_{A_l}(X_t) - \frac{\epsilon}{2^7}.$$

Combining this with Eq (6.11) shows that

$$\mathbb{E}\left[\max_{T_p^\star\leq T<T_{p+1}^\star}\frac{1}{T}\sum_{t=1}^{T} r_t(a_t^\star) - r_t(\hat{a}_t)\right] \geq \mathbb{E}\left[\frac{1}{\hat{T}}\sum_{t=T_p^\star}^{\hat{T}} r_t(a_t^\star) - r_t(\hat{a}_t) - \frac{T_p^\star}{\hat{T}}\right] \geq \frac{\epsilon}{8} - \frac{\epsilon}{2^{10}} - \frac{\epsilon}{2^7}. \tag{6.12}$$

As a last step before defining new rewards, we introduce the scale $\delta_l > 0$ such that

$$\mathbb{P}\left[\min_{1\leq t,t'<2^{R_{p+1}^\star},X_t\neq X_{t'}} \rho(X_t, X_{t'}) \leq \delta_l\right] \leq \frac{\epsilon}{2^{10}}.$$

We denote by $\mathcal{H}$ the complement event.

We are now ready to introduce the new online rewards. To do so, we first need to introduce some notations for partitions of the space $\mathcal{X}$. Let $(x^u)_{u\geq 1}$ be a dense sequence in $\mathcal{X}$. We define the sets $P_u = (A_l \cap B(x^u, \delta_l)) \setminus \bigcup_{v<u} B(x^u, \delta_l)$ for $u \geq 1$. We can easily check that the sequence of measurable sets $(P_u)_{u\geq 1}$ forms a partition of A_l, and that each set P_u has diameter at most δ_l. For any binary sequence $\boldsymbol{b} = (b_u)_{u\geq 1}$, we define online rewards that follow the same structure as defined with the procedure from Algorithm 6.1, with the difference that rewards $r_t^{\boldsymbol{b}}$, at any active time $t \in \mathcal{S}$ with $X_t \in P_u$ for some $u \geq 1$,

Input: Binary sequence $\boldsymbol{b}$
Let $(B_t)_{t\geq 1}$ be an i.i.d. $\mathcal{B}(\frac{1}{2})$ sequence
for $t = 1, \ldots, T_p^\star - 1$ **do**
  Observe context X_t
  Define $r_t(a) = r_t^\star(a)$ for all $a \in \mathcal{A}$
  Observe action selected by learner $\hat{a}_t$
end
Initialize $i(R_p^\star) = 0$ and let $\mathcal{S}_{T_p^\star - 1} = \emptyset$
for $r = R_p^\star, \ldots, R_{p+1}^\star - 1$ **do**
  for $t = 2^r, \ldots, 2^{r+1} - 1$ **do**
    Observe context X_t
    if $t \notin \mathcal{T}^{i(r)}$ **then**
      Let $r_t^{\boldsymbol{b}}(a) = 0$ for all $a \in \mathcal{A}$ and $\mathcal{S}_t = \mathcal{S}_{t-1}$
    else if $\forall T_p\star \leq t' < t$, $X_{t'} \neq X_t$; $X_t \in P_u$ *for some* $u \geq 1$ **then**
      Let $r_t^{\boldsymbol{b}}(a) = \begin{cases} b_u & a = a_1 \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\} \end{cases}$ for $a \in \mathcal{A}$
      $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{t\}$
    else if $\forall T_p\star \leq t' < t$, $X_{t'} \neq X_t$ **then**
      Let $r_t^{\boldsymbol{b}}(a) = \begin{cases} B_t & a = a_1 \\ \frac{3}{4} & a = a_2, \\ 0 & a \notin \{a_1, a_2\} \end{cases}$ for $a \in \mathcal{A}$
      $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{t\}$
    else if $\exists T_p\star \leq t' < t$ *such that* $X_t = X_{t'}$, $t' \in \mathcal{S}_{t-1}$ *and* $\hat{a}_{t'} = a_1$ **then**
      Let $r_t^{\boldsymbol{b}}(a) = 0$ for all $a \in \mathcal{A}$ and $\mathcal{S}_t = \mathcal{S}_{t-1}$
    else
      Define $r_t^{\boldsymbol{b}}(a) = r_{t'}(a)$ for all $a \in \mathcal{A}$ where $t' < t$, $X_t = X_{t'}$ and $t' \in \mathcal{S}_{t-1}$
      $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{t\}$
    end
    Observe action selected by learner $\hat{a}_t$
    while $\frac{1}{t}\sum_{u=T_p^\star}^{t} \mathbb{1}_{u\in\mathcal{S}_t}\mathbb{1}_{\hat{a}_u \neq a_2} \geq 2^{-2i(r)}$ **do** $i(r) \leftarrow \max(0, i(r) - 1)$;
  end
  Define $i(r+1) = \min\{i(r) + 1, k\}$ where k is such that $Q_p(k) \leq r < Q_p(k+1)$
end

Algorithm 6.2: Procedure to define the online rewards $\boldsymbol{r}^{\boldsymbol{b}}_{<T_{p+1}^\star}$

are constructed using the binary value b_u instead of the random binary variable $B_{u(t)}$ where $u(t) = \min\{T_p^\star \leq u \leq t : X_t = X_u\}$. The procedure to construct the rewards $\boldsymbol{r}^{\boldsymbol{b}}$ until time $T_{p+1}^\star$ is given in Algorithm 6.2.

Consider the case when the binary sequence $\boldsymbol{b}$ is sampled as an i.i.d. $\mathcal{B}(\frac{1}{2})$ process. We argue that under the event $\mathcal{H}$, these rewards $\boldsymbol{r}^{\boldsymbol{b}}$ from Algorithm 6.2 are not distinguishable from the rewards $\boldsymbol{r}$ from Algorithm 6.1. First, observe that they share the same overall

structure, the only difference is that when needed to define rewards $r_t^{\boldsymbol{b}}$ at an active time $t \in \mathcal{S}$, one may use b_u instead of B_t, where u is such that $X_t \in P_u$. Recall that b_u is by hypothesis sampled as $b_u \sim \mathcal{B}(\frac{1}{2})$ as B_t and further, under the event $\mathcal{H}$, all distinct points from $\mathbb{X}_{<T^\star_{p+1}}$ falling within A_l are at distance at least δ_l. We only use b_u for $r_t^{\boldsymbol{b}}$ when $X_t \in P_u$. Therefore, under $\mathcal{H}$, one has $\{t' < t : X_t \in P_u\} = \emptyset$. This shows that the variable b_u was never observed before time t and as a result, is not distinguishable from a true random binary variable $B_t \sim \mathcal{B}(\frac{1}{2})$. In particular, under $\mathcal{H}$, the rewards $\boldsymbol{r}^{\boldsymbol{b}}$ when $\boldsymbol{b} \overset{i.i.d.}{\sim} \mathcal{B}(\frac{1}{2})$, yield the same selected actions as the rewards $\boldsymbol{r}$ from Algorithm 6.1. Now for any binary sequence $\boldsymbol{b}$, we define the policy

$$\pi^{\boldsymbol{b}}(x) = \begin{cases} a_1 & \text{if } b_u^k = 1, x \in P_u, \\ a_2 & \text{if } b_u^k = 0, x \in P_u, \\ a_2 & \text{if } x \notin A_l. \end{cases}$$

By construction, these are constructed exactly similarly to the best action in hindsight $a_t^\star$ for contexts falling in A_l as defined previously. Therefore,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{b} \overset{i.i.d.}{\sim} \mathcal{B}(\frac{1}{2})} \left[\mathbb{E}_{\mathbb{X}, \boldsymbol{a}} \left(\max_{T_p^\star \le T < T_{p+1}^\star} \frac{1}{T} \sum_{t=1}^{T} r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\hat{a}_t) \right) \right] \\
&\quad \ge \mathbb{P}[\mathcal{H}] \cdot \mathbb{E}_{\mathbb{X}|\mathcal{G}} \left[\mathbb{E}_{\boldsymbol{b} \overset{i.i.d.}{\sim} \mathcal{B}(\frac{1}{2}), \boldsymbol{a}} \left(\max_{T_p^\star \le T < T_{p+1}^\star} \frac{1}{T} \sum_{t=1}^{T} r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\hat{a}_t) \right) \mid \mathbb{X}, \mathcal{G} \right] \\
&\quad = \mathbb{P}[\mathcal{H}] \cdot \mathbb{E}_{\mathbb{X}|\mathcal{G}} \left[\mathbb{E}_{\boldsymbol{a}} \left(\max_{T_p^\star \le T < T_{p+1}^\star} \frac{1}{T} \sum_{t=1}^{T} r_t(a_t^\star) - r_t(\hat{a}_t) \right) \mid \mathbb{X}, \mathcal{G} \right] \\
&\quad \ge \mathbb{E}_{\mathbb{X}, \boldsymbol{a}} \left[\max_{T_p^\star \le T < T_{p+1}^\star} \frac{1}{T} \sum_{t=1}^{T} r_t(a_t^\star) - r_t(\hat{a}_t) \right] - \mathbb{P}[\mathcal{H}^c] \ge \frac{\epsilon}{8} - \frac{\epsilon}{2^6}.
\end{aligned}$$

In particular, there exists a realization $\boldsymbol{b}$ such that

$$\mathbb{E} \left[\max_{T_p^\star \le T < T_{p+1}^\star} \frac{1}{T} \sum_{t=T_p^\star}^{T} r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\hat{a}_t) \right] \ge \frac{\epsilon}{8} - \frac{\epsilon}{2^6}. \tag{6.13}$$

We fix this realization of $\boldsymbol{b}$ in the rest of the proof. We are now ready to close the induction by letting $B_{p+1} := A_l \setminus (B_1 \cup \ldots \cup B_p)$ and defining the policy $\pi^{(p+1)}$ so as to be consistent with the selected actions of $\pi^{(p)}$ on $B_1, \ldots, B_p$. We pose

$$\pi^{(p+1)}(x) = \begin{cases} \pi^{(p)} & \text{if } x \in B_1 \cup \ldots \cup B_p, \\ \pi^{\boldsymbol{b}} & \text{otherwise.} \end{cases}$$

Observe that by construction, $\pi^{(p+1)}(x) = a_2$ for all $x \notin B_1 \cup \ldots \cup B_{p+1}$. Next, we define the rewards $r_t^\star$ to be exactly $r_t^{\boldsymbol{b}}$ for any $t < T_{p+1}^\star$. Note that by the construction given in Algorithm 6.2, these rewards are consistent with the rewards $r_t^\star$ that had already been constructed for $t < T_p^\star$. In the rest of the proof, we show that these satisfy the induction requirements.

We first check that the fact that $\pi^{(p+1)}$ differs from $\pi^{(p)}$ on A_l does not affect significantly the guarantees of the constructed rewards until time $T_p^\star$. Indeed, for any $T < T_p^\star$,

$$\left|\sum_{t=1}^{T} r_t^\star(\pi^{(p+1)}) - r_t^\star(\pi^{(p)})\right| \leq |\{t \leq T : X_t \in A_l\}| \leq T\mathbb{1}_{\exists t \leq T : Xt \in A_l},$$

so that, using Eq (6.7) and Eq (6.9), for any $p' \leq p$,

$$\begin{aligned}
&\mathbb{E}\left[\max_{T^\star_{p'-1} \leq T < T^\star_{p'}} \frac{1}{T}\sum_{t=1}^{T} r_t^\star(\pi^{(p+1)}(X_t)) - r_t^\star(\hat{a}_t)\right] \\
&\geq \mathbb{E}\left[\max_{T^\star_{p'-1} \leq T < T^\star_{p'}} \frac{1}{T}\sum_{t=1}^{T} r_t^\star(\pi^{(p+1)}(X_t)) - r_t^\star(\hat{a}_t)\right] - \mathbb{P}(\exists t < T_p^\star : X_t \in A_l) \\
&\geq \frac{\epsilon}{16} + \frac{\epsilon}{2^{p+10}} - \frac{\epsilon}{2^{p+11}} \geq \frac{\epsilon}{16} + \frac{\epsilon}{2^{p+11}}.
\end{aligned}$$

Now we check that the guarantee also holds for $p' = p+1$. First, recall that by construction of Algorithm 6.2, for any $r \geq R_p^\star$, one has that $i(r) \leq k$ where k is such that $Q_p(k) \leq r < Q_p(k+1)$. In particular, the active times during the corresponding period satisfy $\mathcal{S} \cap [2^r, 2^{r+1}) \subset \mathcal{T}^k$. As a result, we obtain $\mathcal{S} \subset \mathcal{T}^{(p)}$, where we recall that $\mathcal{T}^{(p)} := \bigcup_{i \geq 0} \mathcal{T}^i \cap \{t \geq 2^{Q_p(i)}\}$. Then, because $\pi^{(p+1)}$ only differs from $\pi^{\boldsymbol{b}}$ on $B_1 \cup \ldots \cup B_p$, for any $T_p^\star \leq T < T_{p+1}^\star$,

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\pi^{(p+1)}(X_t)) &\leq \frac{1}{T}\sum_{t \leq T, t \in \mathcal{S}} (r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\pi^{(p+1)}(X_t))) \\
&\leq \frac{1}{T}\sum_{t \leq T, t \in \mathcal{T}^{(p)}} \sum_{p'=1}^{p} \mathbb{1}_{B_{p'}}(X_t) \\
&\leq \sum_{p'=1}^{p} \sup_{T \geq 1} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p)}} \mathbb{1}_{B_{p'}}(X_t).
\end{aligned}$$

Therefore, combining Eq (6.13) and the induction hypothesis Eq (6.8), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\max_{T_p^\star \leq T < T_{p+1}^\star} \frac{1}{T}\sum_{t=1}^{T} r_t^{\boldsymbol{b}}(\pi^{(p+1)}(X_t)) - r_t^{\boldsymbol{b}}(\hat{a}_t)\right] \\
&\geq \mathbb{E}\left[\max_{T_p^\star \leq T < T_{p+1}^\star} \frac{1}{T}\sum_{t=1}^{T} r_t^{\boldsymbol{b}}(\pi^{\boldsymbol{b}}(X_t)) - r_t^{\boldsymbol{b}}(\hat{a}_t)\right] - \sum_{p'=1}^{p} \mathbb{E}\left[\sup_{T \geq 1} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p)}} \mathbb{1}_{B_{p'}}(X_t)\right] \\
&\geq \frac{\epsilon}{8} - \frac{\epsilon}{2^6} - \frac{\epsilon}{2^{10}} \geq \frac{\epsilon}{16} + \frac{\epsilon}{2^{p+10}}.
\end{aligned}$$

The last step consists of constructing the increasing indices $Q_{p+1}(i)$ for $i \geq 0$. By the dominated convergence theorem, for any $i \geq 0$, there exists $\tilde{T}_i \geq 1$ such that

$$\mathbb{E}\left[\sup_{T \geq \tilde{T}_i} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^i} \mathbb{1}_{B_{p+1}}(X_t) - \hat{\mu}_{(X_t)_{t \in \mathcal{T}^i}}(B_{p+1})\right] \leq \frac{\epsilon}{2^{p+12+i}}.$$

We then define by induction the sequence of integers $Q_{p+1}(i)$ that satisfy the following two properties. First, $Q_{p+1}(0) \geq \max(Q_p(0), \log_2 \tilde{T}_0)$, second, for all $i \geq 1$, $Q_{p+1}(i) \geq \max(Q_p(i), \log_2 \tilde{T}_i, Q_{p+1}(i-1))$. In particular, the sequence is increasing and the above equation shows that

$$\mathbb{E}\left[\sup_{T \geq 2^{Q_{p+1}(i)}} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^i} \mathbb{1}_{B_{p+1}}(X_t) - \hat{\mu}_{(X_t)_{t \in \mathcal{T}^i}}(B_{p+1})\right] \leq \frac{\epsilon}{2^{p+12+i}}. \tag{6.14}$$

Now letting $\mathcal{T}^{(p+1)} = \bigcup_{i \geq 0} \mathcal{T}^i \cap \{t \geq 2^{Q_{p+1}(i)}\}$, we note that

$$\begin{aligned}
\sup_{T \geq 1} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p+1)}} \mathbb{1}_{B_{p+1}}(X_t) &= \sup_{i \geq 0} \sup_{2^{Q_{p+1}(i)} \leq T < 2^{Q_{p+1}(i+1)}} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p+1)}} \mathbb{1}_{B_{p+1}}(X_t) \\
&\leq \sup_{i \geq 0} \sup_{2^{Q_{p+1}(i)} \leq T < 2^{Q_{p+1}(i+1)}} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^i} \mathbb{1}_{B_{p+1}}(X_t).
\end{aligned}$$

As a result,

$$\begin{aligned}
&\mathbb{E}\left[\sup_{T \geq 1} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p+1)}} \mathbb{1}_{B_{p+1}}(X_t)\right] \\
&\leq \mathbb{E}\left[\sup_{i \geq 0} \hat{\mu}_{(X_t)_{t \in \mathcal{T}^i}}(B_{p+1})\right] + \sum_{i \geq 0} \mathbb{E}\left[\sup_{T \geq 2^{Q_{p+1}(i)}} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^i} \mathbb{1}_{B_{p+1}}(X_t) - \hat{\mu}_{(X_t)_{t \in \mathcal{T}^i}}(B_{p+1})\right] \\
&\leq \sup_{i \geq 0} \mathbb{E}\left[\hat{\mu}_{(X_t)_{t \in \mathcal{T}^i}}(B_{p+1})\right] + \frac{\epsilon}{2^{p+11}} \leq \frac{\epsilon}{2^{p+10}}.
\end{aligned}$$

In the second inequality we used Eq (6.14), and in the third inequality, we used Eq (6.9). Finally, because for all $i \geq 0$, one has $Q_{p+1}(i) \geq Q_p(i)$, we have directly $\mathcal{T}^{(p)} \subset \mathcal{T}^{(p+1)}$, which shows that for all $p' \leq p$, we still have

$$\mathbb{E}\left[\sup_{T \geq 1} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}^{(p+1)}} \mathbb{1}_{B_{p'}}(X_t)\right] \leq \frac{\epsilon}{2^{p'+10}}, \quad p' \leq p.$$

This ends the inductive construction of the rewards $\boldsymbol{r}^\star$.

The last step of the proof is to show that $f_\cdot$ is not universally consistent under $\mathbb{X}$ for these online rewards $\boldsymbol{r}^\star$. Having constructed the sequence of sets $(B_p)_{p \geq 1}$, we let $\pi^\star$ be the policy defined by

$$\pi^\star(x) = \begin{cases} \pi^{(p)}(x) & \text{if } x \in B_p, \\ a_2 & \text{otherwise.} \end{cases}$$

Recall that the sequence of policies $\pi^{(p)}$ for $p \geq 1$ was constructed so that they are consistent: $\pi^{(p')}$ for $p' \geq p \geq 1$ all coincide on A_p. Further, all $\pi^{(p)}$ coincide on $(\bigcup_{p \geq 1} B_p)^c$ on which they select a_2. Now fix $p \geq 1$. Because the rewards are also constructed to be consistent over

time, if $\hat{a}_t$ denotes the selected action at time t for rewards $\boldsymbol{r}^\star$, the induction implies that for all $p' \geq p$ one has

$$\mathbb{E}\left[\max_{T^\star_{p-1} \leq T < T^\star_p} \frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^{(p')}(X_t)) - r^\star_t(\hat{a}_t)\right] \geq \frac{\epsilon}{16}. \tag{6.15}$$

As a result, because $\pi^{(p')}$ and $\pi^\star$ coincide everywhere except on $\bigcup_{q>p'} B_q$, we have for any $T^\star_{p-1} \leq T < T^\star_p$,

$$\frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^\star(X_t)) - r^\star_t(\hat{a}_t) \geq \frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^{(p')}(X_t)) - r^\star_t(\hat{a}_t) - \mathbb{1}\left(\exists t < t^\star_p : X_t \in \textstyle\bigcup_{q>p'} B_q\right).$$

Because the sets $(B_p)_{p\geq 1}$ are all disjoint, we have $\mathbb{P}\left(\exists t < t^\star_p : X_t \in \bigcup_{q>p'} B_q\right) \to 0$ as $p' \to \infty$. Thus, using Eq (6.15) yields

$$\mathbb{E}\left[\max_{T^\star_{p-1} \leq T < T^\star_p} \frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^\star(X_t)) - r^\star_t(\hat{a}_t)\right] \geq \frac{\epsilon}{16}.$$

Because this holds for all $p \geq 1$, Fatou's lemma implies

$$\begin{aligned}\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^\star(X_t)) - r^\star_t(\hat{a}_t)\right] &\geq \limsup_{p\to\infty} \mathbb{E}\left[\max_{T^\star_{p-1} \leq T < T^\star_p} \frac{1}{T}\sum_{t=1}^{T} r^\star_t(\pi^\star(X_t)) - r^\star_t(\hat{a}_t)\right] \\ &\geq \frac{\epsilon}{16}.\end{aligned}$$

As a result, the learning rule is not universally consistent under $\mathbb{X}$, which ends the proof of the theorem. ■

6.5.2 A sufficient condition on learnable processes

In this section, we show that $\mathcal{C}_5$ is sufficient universal learning for all reward models. We recall that the condition $\mathcal{C}_5$ asks that there exists an increasing sequence $(T_i)_{i\geq 0}$ such that $\mathbb{X}^{\mathcal{T}} \in \mathrm{CS}$ where $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$ is obtained by adding the times $\mathcal{T}^i$ according to the rate given by $(T_i)_{i\geq 0}$.

It is straightforward to see $\mathrm{CS} \subset \mathcal{C}_5$ since for any $\mathbb{X} \in \mathrm{CS}$, one can take any arbitrary sequence, for instance, $T_i = i$ for $i \geq 0$, and satisfy property $\mathcal{C}_5$. Before showing that $\mathcal{C}_5$ is a sufficient condition for universal learning with online rewards, we state a known result showing that for CS processes, there is a countable sequence of policies that is empirically dense within all measurable policies.

Lemma 6.4 ([Han21a] Lemma 24)**.** *Let $\mathcal{A}$ be a countable action space and $\mathcal{X}$ a separable metrizable Borel space. There exists a countable sequence of measurable policies $(\pi^l)_{l\geq 1}$ from $\mathcal{X}$ to $\mathcal{A}$ such that for any extended process $\mathbb{X}^{\mathcal{T}} \in CS$, and any measurable policy $\pi : \mathcal{X} \to \mathcal{A}$,*

$$\inf_{l\geq 1} \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}[\pi^l(X_t) \neq \pi(X_t)]\right] = 0.$$

We are now ready to prove the sufficiency of $\mathcal{C}_5$.

Theorem 6.12. *Let $\mathcal{X}$ be a metrizable separable Borel space and $\mathcal{A}$ a finite action space. Then, $\mathcal{C}_5 \subset \textsc{SOAB}_{online}$.*

Proof Let $\mathbb{X} \in \mathcal{C}_5$, and $(T_i)_{i\geq 0}$ such that letting $\mathcal{T} = \bigcup_{i\geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$ we have $\mathbb{X}^{\mathcal{T}} \in \text{CS}$. We suppose that $T_i = 2^{u(i)}$ for some indices $u(i)$ increasing in i. This is without loss of generality because one could take $\tilde{T}_i = \min\{2^s, 2^s \geq T_i\}$ and still have an extended CS process in the definition of $\mathbb{X}^{\mathcal{T}}$ (a slower sequence $(T_i)_i$ only reduces considered points, hence does not impact the CS property). We may also suppose that $u(i) \geq 2i$. Also, letting $\eta_i = \sqrt{\frac{8\ln(i+1)}{2^i}}$ for $i \geq 0$, we suppose that $u(i) \geq \eta_i 2^{i+5}$. Last, we suppose that $u(0) = 0$ which again can be done without loss of generality since the CS property is not affected by the behavior of the process on the first T_0 times. Hence, $T_0 = 1$.

Similarly to the algorithm that we proposed for stationary rewards in Chapter 5, the learning rule associates a category p to each time t and acts separately on each category. To do so, the algorithm first computes the phase of t as follows: $\textsc{Phase}(t)$ is the unique integer i such that $T_i \leq t < T_{i+1}$. Then, we define the stage $\textsc{Stage}(t) := \lfloor \log_2 t \rfloor = l$ so that $t \in [2^l, 2^{l+1})$, and the period $k = \textsc{Period}(t)$ as the unique integer k such that $T_i^{l2^i+k} \leq t < T_i^{l2^i+k+1}$ where $i = \textsc{Phase}(t)$. (Recall that $T_i^{l2^i} = 2^l$). We will refer to $[T_i^{l2^i+k}, T_i^{l2^i+k+1})$ as period k of stage l of phase i. The category of t is then defined in terms of the number of occurrences of X_t within its period.

$$\textsc{Category}(t, \mathbb{X}_{\leq t}) := \left\lfloor \log_4 \sum_{t'=T_i^{l2^i+k}}^{t} \mathbb{1}[X_{t'} = X_t] \right\rfloor,$$

where $i = \textsc{Phase}(t)$, $l = \textsc{Stage}(t)$, $k = \textsc{Period}(t)$. For conciseness, we will omit the argument $\mathbb{X}_{\leq t}$ of the function in the rest of the proof. In words, category p contains duplicates with indices in $[4^p, 4^{p+1})$ within the periods defined by $\mathcal{T}$. Now using Lemma 6.4, let $(\pi^l)_{l\geq 1}$ be a sequence of dense functions from $\mathcal{X}$ to $\mathcal{A}$ within measurable functions under extended CS processes. The learning rule acts separately on times from different categories. We now fix a category p and only consider points from this category. Essentially, between times T_i and T_{i+1}, the learning rule performs the Hedge algorithm for learning with experts to select between the strategies j for $1 \leq j \leq i$, which apply π^j and a strategy 0 which assigns a different EXP3.IX learner to each new instance within each period at scale i.

Precisely, during an initial phase $[1, 2^{u(16p)})$, the learning rule only applies strategy 0. Then, let $l \geq u(16p)$ and $u(i) \leq l < u(i+1)$, we define the learning rule on stage $[2^l, 2^{l+1})$ as follows. For $0 \leq k < 2^i$, before period k of stage l, we construct probabilities $P_p(l, k; j)$ for $j = 0, \ldots, i$. These will be probabilities of exploration for each strategy. At the first phase $k = 0$ we initialize at the uniform distribution $P_p(l, 0; j) = \frac{1}{i+1}$. During period k, each new time of category p is assigned a strategy $\hat{j}(t)$ sampled independently from the past according to probabilities $P_p(l, k; \cdot)$. Duplicates of X_t within the same category and period are also assigned the same strategy $\hat{j}(t)$. The learning rule then performs the assigned strategy: for $\hat{j} = 0$, it performs an EXP3.IX algorithm and for $1 \leq \hat{j} \leq i$, it applies the policy $\pi^{\hat{j}}$. At the

$\eta_i = \sqrt{\frac{8\ln(i+1)}{2^i}}, i \geq 0$ // learning rates for Hedge

$\hat{r}_p^j(l,0) = 0, P_p(l,0;j) = \frac{1}{i+1}, \quad p,l,j \geq 0$ // initialization

for $t \geq 1$ **do**

  Observe context X_t

  $i = \textsc{Phase}(t)$, $l = \textsc{Stage}(t)$, $k = \textsc{Period}(t)$, $p = \textsc{Category}(t)$,

  $S_t = \{t' \in [T_i^{l2^i+k}, t) : \textsc{Category}(t') = p, X_{t'} = X_t\}$

  if $t < 2^{u(16p)}$ **then** // initially play strategy 0

    $\hat{a}_t = \text{EXP3.IX}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$

  else

    if $S_t = \emptyset$ **then** $\hat{j}(t) \sim P_p(l,k;\cdot)$ // select strategy $\hat{j}(t)$

    else $\hat{j}(t) = \hat{j}(\min S_t)$

    if $\hat{j}(t) = 0$ **then** $\hat{a}_t = \text{EXP3.IX}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$ // play strategy $\hat{j}(t)$

    else $\hat{a}_t = \pi^{\hat{j}(t)}(X_t)$

  end

  Receive reward r_t

  if $l \geq u(16p)$, $t = T_i^{l2^i+k+1} - 1$ **then** // update probabilities

    $\hat{r}_p(l,k+1;j) = \hat{r}(l,k;j) + \frac{1}{2^{l-i}} \sum_{t \in [T_i^{l2^i+k}, T_i^{l2^i+k+1})} \frac{\mathbb{1}[\textsc{Category}(t)=p, \hat{j}(t)=j]}{P_p(l,k;j)} r_t, \quad 0 \leq j \leq i$

    $P_p(l,k+1;j) = \frac{\exp(\eta_i \hat{r}_p(l,k+1;j))}{\sum_{j'=0}^{i} \exp(\eta_i \hat{r}_p(l,k+1;j'))}, \quad 0 \leq j \leq i$

  end

end

Algorithm 6.3: Learning rule for $\mathcal{C}_5$ processes on times $\mathcal{T}_p$

end of the phase, the learning rule computes the average reward obtained by each strategy,

$$\tilde{r}_p(l,k;j) := \frac{1}{2^{l-i}} \sum_{T_i^{l2^i+k} \leq t < T_i^{l2^i+k+1}} \frac{\mathbb{1}[\textsc{Category}(t) = p,\ \hat{j}(t) = j]}{P_p(l,k;j)} r_t,$$

and $\hat{r}_p(l,k+1;j) = \sum_{0 \leq k' \leq k} \tilde{r}_p(l,k';j)$ the cumulative average reward of strategy j. These rewards are then used to define the probabilities for the next phase $P_p(l,k+1;\cdot)$ using the exponentially weighted averages.

$$P_p(l,k+1;j) = \frac{\exp(\eta_i \hat{r}_p(l,k+1;j))}{\sum_{j'=0}^{i} \exp(\eta_i \hat{r}_p(l,k+1;j'))},$$

where $\eta_i = \sqrt{\frac{8\ln(i+1)}{2^i}}$ is the parameter of the Hedge algorithm for 2^i steps. The detailed algorithm is given in Algorithm 6.3.

We now show that this is a universally consistent algorithm for $\mathbb{X}$. We first introduce some notations. For $p \geq 0$,

$$\mathcal{T}_p := \bigcup_{i \geq 1} [T_i, T_{i+1}) \cap \left\{ t \geq 1 : T_i^k \leq t < T_i^{k+1}, 4^p \leq \sum_{t'=T_i^k}^{t} \mathbb{1}[X_{t'} = X_t] < 4^{p+1} \right\},$$

is the set of times in category p. We will also denote $\mathbb{X}^p := (X_t)_{t\in\mathcal{T}^p}$. In this setting, the rewards are independent of the selected actions of the learner. First, note that the constructed rewards $\hat{r}_p(l,k;j)$ are estimates of the average reward that would have been obtained by strategy j during period k of stage l. For convenience, we denote $\mathcal{T}_p(k,l) = [T^{l2^i+k}, T^{l2^i+k+1}) \cap \mathcal{T}_p$. We denote by $R_p(l,k;j)$ the reward that would have been obtained had we selected always $\hat{j} = j$ on this period, and $r_p(l,k;j) = \frac{R_p(l,k;j)}{2^{l-i}}$ the average reward of strategy j for $0 \le j \le i$. For example, for strategy $1 \le j \le i$ we have $R_p(l,k;j) = \sum_{t\in\mathcal{T}_p(l,k)} r_t(\pi^j(X_t))$. Let $\mathcal{X}_p(l,k) = \{X_t, t \in \mathcal{T}_p(k,l)\}$ the set of visited instances during this period. For $x \in \mathcal{X}_p(l,k)$ we denote $t_p(l,k;x) = \min\{t \in \mathcal{T}_p(k,l) : X_t = x\}$ the first time of occurrence of x during this period, and $N_p(l,k;x) = |\{t \in \mathcal{T}_p(l,k) : X_t = x\}|$ its number of occurrences. Let $0 \le j \le i$. We use Hoeffding's inequality conditionally on $\mathbb{X}$ and $P_p(l,k;j)$, to obtain

$$\mathbb{P}\left[\left|\sum_{x\in\mathcal{X}_p(l,k)} \mathbb{1}[\hat{j}(t) = j] \sum_{t\in\mathcal{T}_p(l,k), X_t = x} r_t - P_p(l,k;j)R_p(l,k;j)\right| \right.$$
$$\left. \ge P_p(l,k;j)4^{p+1}2^{\frac{3}{4}(l-i)} \mid \mathbb{X}, P_p(l,k;j)\right]$$
$$\le 2\exp\left(-2\frac{P_p(l,k;j)^2 2^{3/2(l-i)}}{|\mathcal{X}_p(l,k)|}\right) \le 2\exp\left(-2\frac{2^{3/2(l-i)}}{(i+1)^2 e^{\eta_i 2^{i+1}}|\mathcal{X}_p(l,k)|}\right).$$

Now by construction of $\mathcal{T}_p(l,k)$, each instance of $\mathcal{X}_p(l,k)$ has at least 4^p duplicates within the same period. Hence $|\mathcal{X}_p(l,k)| \le \frac{2^{l-i}}{4^p}$. As a result, dividing the inner inequality by $P_p(l,k;j)2^{l-i}$, we obtain for $l \ge u(16p)$, with probability at least $1 - 2\exp(-\frac{2^{2p+(l-i)/2}}{(i+1)^2 e^{\eta_i 2^{i+1}}}) := 1 - p_1(l,k;p)$,

$$|\hat{r}_p(l,k;j) - r_p(l,k;j)| < \frac{4^{p+1}}{2^{(l-i)/4}} \le \frac{4}{2^{l/16}}, \tag{6.16}$$

where in the last inequality we used $l \ge u(i) \ge 2i$ and $l \ge u(16p) \ge 32p$. We now focus on the rewards for strategy 0. For any $t \in \mathcal{T}_p(l,k)$ we denote by $\tilde{r}_t$ the reward that would have been obtained had we selected strategy 0 for time t, i.e. $\hat{j}(t_p(l,k;X_t)) = 0$. In particular, we have $R_p(l,k;0) = \sum_{t\in\mathcal{T}_p(l,k)} \tilde{r}_t$. Let $\pi^* : \mathcal{X} \to \mathcal{A}$ be a measurable policy, we now compare $R_p(l,k;0)$ to the rewards obtained by the policy π^* on $\mathcal{T}_p(l,k)$. Intuitively, we wish to apply Theorem 5.6 independently for each EXP3.IX algorithm corresponding to elements of $\mathcal{X}_p(l,k)$. However, these runs are not independent for general adaptive adversaries. Therefore, we will need to go back to the standard analysis of EXP3.IX. Using the same notations as in this analysis, for $t \in \mathcal{T}_p(l,k)$, denote $u(t) = |\{t' \le t : t' \in \mathcal{T}_p(l,k), X_{t'} = X_t\}|$ the index of t for its corresponding EXP3.IX learner. Let $\eta_u = 2\gamma_u = \sqrt{\frac{\ln|\mathcal{A}|}{u|\mathcal{A}|}}$ be the parameters used by the learner at step u. Also, denote by $p_{t,a}$ the probability that the EXP3.IX learner chose $a \in \mathcal{A}$ at time t. Further, for $a \in \mathcal{A}$ denote by $\ell_{t,a} = 1 - r_t(a)$ and $\tilde{\ell}_{t,a} = \frac{\ell_{t,a}}{p_{t,a}+\gamma_{u(t)}}\mathbb{1}[a \text{ selected}]$. We keep in mind that the term "selected" refers to the selection of the EXP3.IX algorithm, but not necessarily the selection of our learning rule, which potentially did not apply strategy 0 at that time. To avoid confusion, for $t \in \mathcal{T}_p(l,k)$, denote $\tilde{a}_t$ the action that would be selected

by the EXP3.IX learner at time t. Last, we define

$$A_p(l,k) = \sum_{t \in \mathcal{T}_p(l,k)} \tilde{\ell}_{t,\pi^*(X_t)} - \ell_{t,\pi^*(X_t)} \quad \text{and} \quad B_p(l,k) = \sum_{t \in \mathcal{T}_p(l,k)} \sum_{a \in \mathcal{A}} \eta_{u(t)}(\tilde{\ell}_{t,a} - \ell_{t,a}).$$

Then, the same arguments as in Proposition 6.3 give

$$\begin{aligned}
\sum_{t \in \mathcal{T}_p(l,k)} r_t(\pi^*(X_t)) - r_t(\tilde{a}_t) &\le A_p(l,k) + B_p(l,k) + \sum_{x \in \mathcal{X}_p(l,k)} 3\sqrt{|\mathcal{A}| \ln |\mathcal{A}| N_p(l,k;x)} \\
&\le A_p(l,k) + B_p(l,k) + 3\sqrt{|\mathcal{A}| \ln |\mathcal{A}| 4^{p+1}} |\mathcal{X}_p(l,k)| \\
&\le A_p(l,k) + B_p(l,k) + 6\sqrt{|\mathcal{A}| \ln |\mathcal{A}|} 2^{-p} 2^{l-i},
\end{aligned}$$

where in the last inequality, we used the fact that $|\mathcal{X}_p(l,k)| \le \frac{2^{l-i}}{4^p}$. Now similarly to Proposition 6.3, note that conditionally on $\mathbb{X}$, the increments of $A_p(l,k)$ and $B_p(l,k)$ form a super-martingale with increments upper bounded by $2\sqrt{\frac{|\mathcal{A}|4^{p+1}}{\ln |\mathcal{A}|}}$ and $2|\mathcal{A}|\sqrt{\frac{|\mathcal{A}|4^{p+1}}{\ln |\mathcal{A}|}}$ respectively. Thus, Azuma's inequality implies

$$\begin{aligned}
\mathbb{P}[A_p(l,k) \le 8p|\mathcal{A}| 2^{p+\frac{3}{4}(l-i)} \mid \mathbb{X}] &\ge 1 - e^{-2p^2 2^{(l-i)/2}}, \\
\mathbb{P}[B_p(l,k) \le 8p|\mathcal{A}|^2 2^{p+\frac{3}{4}(l-i)} \mid \mathbb{X}] &\ge 1 - e^{-2p^2 2^{(l-i)/2}}.
\end{aligned}$$

Thus, denoting $\delta_p = 6\frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{2^p}$, for any $l \ge 2i, u(16p)$, with probability at least $1 - 2e^{-2p^2 2^{(l-i)/2}} := 1 - p_2(l,k;p)$, we have

$$R_p(l,k;0) \ge \sum_{t \in \mathcal{T}_p(l,k)} r_t(\pi^*(X_t)) - 16|\mathcal{A}|^2 2^{-i} 2^{15l/16} - \delta_p 2^{l-i}. \tag{6.17}$$

In the first phase where $l < u(16p)$, we will need to proceed differently. Let $\mathcal{T}^{init} = \bigcup_{p \ge 0} \{t \in \mathcal{T}_p : t < 2^{u(16p)}\}$. Observe that in these times, the learning uses a distinct EXP3.IX learner for each new instance within each category and period. In Proposition 6.3 we showed that this learning rule is universally consistent under processes visiting a sublinear number of distinct instances almost surely. We now show that this is the case for the process $(X_t)_{t \in \mathcal{T}^{init}}$ where for any $t, t' \in \mathcal{T}^{init}$, we view X_t and $X_{t'}$ as duplicates if and only if $X_t = X_{t'}$ and they have same category and period. For $l \ge 1$, let $p(l)$ denote the index p such that $u(16p) \le l < u(16(p+1))$ and $i(l)$ be the index i such that $u(i) \le l < u(i+1)$. Fix $T \ge 1$ and let $l \ge 0$ such that $2^l \le T < 2^{l+1}$. We now count the number of distinct instances $N(T)$

of $(X_t)_{t\in\mathcal{T}^{init}}$ before time T. To do so, we distinguish whether $t \le 2^{l/2}$ or $t > 2^{l/2}$ as follows,

$$\begin{aligned}
N(T) \le \sum_{p\ge 0}\sum_{l'\le u(16p),l}\sum_{k}|\mathcal{X}_p(l',k)| &\le 2^{l/2} + \sum_{p\ge p(\frac{l}{2})}\sum_{\frac{l}{2}\le l'\le l}\sum_{k}|\mathcal{X}_p(l',k)| \\
&\le 2^{l/2} + \sum_{p\ge p(\frac{l}{2})}\sum_{\frac{l}{2}\le l'\le l}\sum_{k}\frac{2^{l'-i(l')}}{4^p} \\
&\le 2^{l/2} + \sum_{p\ge p(\frac{l}{2})}\frac{2^{l+1}}{4^p} \\
&\le 2^{l/2} + \frac{2^{l+1}}{4^{p(l/2)-1}} \\
&\le \sqrt{T} + \frac{8T}{4^{p(\log_4(T))}} = o(T).
\end{aligned}$$

Now let $\pi^* : \mathcal{X} \to \mathcal{A}$ be a measurable policy. Because of the above estimate, Proposition 6.3 implies that on an event $\mathcal{E}$ of probability one,

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t\le T, t\in\mathcal{T}^{init}} r_t(\pi^*(X_t)) - r_t \le 0.$$

Now recall that $l \ge u(i) \ge 2i, \eta_i 2^{i+5}$, hence $\frac{2^{(l-i)/2}}{e^{\eta_i 2^{i+1}}} \ge 2^{l/4-\eta_i 2^{i+2}} \ge 2^{l/8}$. As a result,

$$\sum_{p\ge 0}\sum_{l\ge 32p}\sum_{k}(i+1)p_1(l,k;p) + p_2(l,k;p) < \infty.$$

Then, the Borel-Cantelli lemma implies that on an event $\mathcal{F}$ of probability one, there exists $\hat{l}$ such that for all $p \ge 0$, $l \ge \max(\hat{l}, u(16p))$ Eq (6.16) holds, for all $p \ge 0$ and $l \ge \hat{l}$, Eq (6.17) holds, and $\mathcal{E}$ is satisfied. We suppose that this event is met in the rest of the proof.

The probabilities $P_p(l,k;j)$ are chosen according to the Hedge algorithm. As a result, we have that for any $l \ge \max(\hat{l}, u(16p))$, $0 \le k < 2^i$,

$$\max_{0\le j\le i}\sum_{k'\le k}\hat{r}_p(l,k;j) - \sum_{k'\le k}\sum_{j=0}^{i}P_p(l,k;j)\hat{r}_p(l,k;j) \le \frac{\ln(i+1)}{\eta_i} + \frac{(k+1)\eta_i}{8}.$$

We then use Eq (6.16) and $k+1 \le 2^i$ to obtain

$$\max_{0\le j\le i}\sum_{k'\le k} r_p(l,k;j) - \sum_{k'\le k}\sum_{j=0}^{i}P_p(l,k;j)\hat{r}_p(l,k;j) \le 2^i\frac{4}{2^{l/16}} + \frac{\eta_i}{4}2^i$$

As a result,

$$\max_{0\le j\le i}\sum_{k'\le k} R_p(l,k;j) - \sum_{k'\le k}\sum_{t\in\mathcal{T}_p(l,k)} r_t \le 4\cdot 2^{15l/16} + \frac{\eta_i}{4}2^l. \tag{6.18}$$

Now because $l \geq u(16p)$, we have $i \geq 16p$, we have

$$\sum_{0\leq k'\leq k}\sum_{t\in\mathcal{T}_p(l,k')} r_t \geq \sum_{0\leq k'\leq k} R_p(l,k';0) - 4\cdot 2^{15l/16} - \frac{\eta_{16p}}{4}2^l$$
$$\geq \sum_{0\leq k'\leq k}\sum_{t\in\mathcal{T}_p(l,k)} r_t(\pi^*(X_t)) - 20|\mathcal{A}|^2 2^{15l/16} - \left(\delta_p + \frac{\eta_{16p}}{4}\right)2^l,$$

where in the second inequality we used Eq (6.17). Therefore, summing these equations, for any $T \geq 2^{\hat{l}}, 2^{u(16p)}$,

$$\sum_{2^{u(16p)}<t\leq T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)) - r_t \leq 2^{\hat{l}} + c|\mathcal{A}|^2 T^{15/16} + 2\left(\delta_p + \frac{\eta_{16p}}{4}\right)T, \tag{6.19}$$

where $c = \frac{20}{1-2^{-15/16}}$. An important remark is that $\sum_{p\geq 0}(\delta_p + \frac{\eta_{16p}}{4}) < \infty$, which will allow us to consider only a finite number of $p \geq 0$ when comparing the performance of the learning rule compared to π^*.

Before doing so, we show that for all $p \geq 0$, we have $\mathbb{X}^p = \mathbb{X}^{\mathcal{T}_p} \in$ CS. By definition, letting $\mathcal{T} = \bigcup_{i\geq 0}\mathcal{T}^i \cap \{t \geq T_i\}$, we have that $\mathbb{X}^{\mathcal{T}} =\in$ CS. Then note that each instance of $[T_i^k, T_i^{k+1}) \cap \mathcal{T}_p$ has at least one duplicate in $[T_i^k, T_i^{k+1}) \cap \mathcal{T}$ and to each instance of $[T_i^k, T_i^{k+1}) \cap \mathcal{T}$ corresponds at most 4^{p+1} duplicates in $[T_i^k, T_i^{k+1}) \cap \mathcal{T}_p$. As a result, for any set $A \in \mathcal{B}$, we have $\hat{\mu}_{\mathbb{X}^p}(A) \leq 4^{p+1}\hat{\mu}_{\mathbb{X}^\mathcal{T}}(A)$, which yields $\mathbb{E}[\hat{\mu}_{\mathbb{X}^p}(A)] \leq 4^{p+1}\mathbb{E}[\hat{\mu}_{\mathbb{X}^\mathcal{T}}(A)]$. Using the definition of extended CS processes ends the proof that $\mathbb{X}^p \in$ CS for all $p \geq 0$.

Now let $\epsilon > 0$ and p_0 such that $\sum_{p\geq p_0}(\delta_p + \frac{\eta_{16p}}{4}) < \epsilon$. Recall that if $t \in \mathcal{T}_p$, we have $t \geq 4^p$. Therefore, summing Eq (6.19) gives

$$\sum_{p\geq p_0}\sum_{2^{u(16p)}\leq t<T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)) - r_t \leq \sum_{p_0\leq p\leq \log_4 T}\sum_{2^{u(16p)}\leq t<T, t\in\mathcal{T}_p} r_t(\pi(X_t)) - r_t$$
$$\leq 2^{\hat{l}}\log_4 T + c|\mathcal{A}|^2 T^{15/16}\log_4 T + \epsilon T.$$

We now treat the case of $p < p_0$. Because $\mathbb{X}^p \in$ CS, by Lemma 6.4, there exists $r^p \geq 1$ such that

$$\mathbb{E}\left[\limsup_{T\to\infty}\frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}_p}\mathbb{1}[\pi^*(X_t) \neq \pi^{r_p}(X_t)]\right] \leq \frac{\epsilon^2}{2p_0^2}.$$

By dominated convergence theorem, let l^p such that

$$\mathbb{E}\left[\sup_{T\geq 2^{l^p}}\frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}_p}\mathbb{1}[\pi^*(X_t) \neq \pi^{r_p}(X_t)]\right] \leq \frac{\epsilon^2}{p_0^2}.$$

Using the Markov inequality, we have

$$\mathbb{P}\left[\sup_{T\geq 2^{l^p}}\frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}_p}\mathbb{1}[\pi^*(X_t) \neq \pi^{r_p}(X_t)] \geq \frac{\epsilon}{p_0}\right] \leq \frac{\epsilon}{p_0}.$$

By union bound, on an event $\mathcal{G}$ of probability at least $1-\epsilon$, for all $p < p_0$ and $T \geq 2^{l^p}$, we have $\sum_{t\leq T, t\in\mathcal{T}_p} \mathbb{1}[\pi(X_t) \neq \pi^{r_p}(X_t)] < \frac{\epsilon}{p_0}T$. Next, let $l_0 = \max(u(r^p), l^p, p < p_0)$. Thus, any phase $l \geq l_0$, has $r^p \leq i$ for all $p < p_0$. Last, let i_0 such that $\eta_{i_0} \leq 2\frac{\epsilon}{p_0}$. On the event $\mathcal{E}\cap\mathcal{F}\cap\mathcal{G}$, for $p < p_0$, for any $l \geq \hat{l}_1 := \max(l_0, 32p_0, u(i_0), \hat{l})$ and $0 \leq k < 2^i$, Eq (6.18) yields

$$\sum_{2^l \leq t < T_i^{l2^i+k+1}, t\in\mathcal{T}_p} r_t(\pi^{r^p}(X_t)) - r_t \leq 4\cdot 2^{15l/16} + \frac{\eta_i}{4}2^l \leq 4\cdot 2^{15l/16} + \frac{\epsilon}{2p_0}2^l$$

As a result, for $T \geq 1$, letting $i(T), l(T)$ the indices i, l such that $2^{u(i)} \leq T < 2^{u(i+1)}$ and $2^l \leq T < 2^{l+1}$, on $\mathcal{E}\cap\mathcal{F}\cap\mathcal{G}$,

$$\begin{aligned}\sum_{p<p_0}\sum_{2^{u(16p)}\leq t\leq T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)) - r_t &\leq 2^{\hat{l}_1} + 2^{-i(T)}T + \sum_{p<p_0}\sum_{t<T, t\in\mathcal{T}_p}\mathbb{1}[\pi^*(X_t)\neq\pi^{r_p}(X_t)] \\ &\quad + \sum_{p<p_0}\sum_{\hat{l}_1\leq l'\leq l}\left(4\cdot 2^{15l'/16} + \frac{\epsilon}{2p_0}2^{l'}\right) \\ &\leq 2^{\hat{l}_1} + 2^{-i(T)}T + \epsilon T + cp_0T^{15/16} + \epsilon T.\end{aligned}$$

Finally, putting everything together, for T sufficiently large, we have

$$\begin{aligned}\sum_{t\leq T} r_t(\pi^*(X_t)) - r_t &\leq \sum_{t\in\mathcal{T}^{init}, t\leq T}\bar{r}_t(\pi^*(X_t)) - r_t + \sum_{p\geq 0}\sum_{2^{u(16p)}\leq t\leq T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)) - r_t \\ &\leq 2^{\hat{l}_1+1}\log_4 T + 2^{-i(T)}T + c(|\mathcal{A}|^2 + p_0)T^{15/16}\log_4 T + 3\epsilon T + \sum_{t\in\mathcal{T}^{init}, t\leq T} r_t(\pi^*(X_t)) - r_t,\end{aligned}$$

which shows that on $\mathcal{E}\cap\mathcal{F}\cap\mathcal{G}$,

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t \leq 3\epsilon.$$

We denote by $(x)_+ = \max(0, x)$ the positive part. Recall that $\mathbb{P}[\mathcal{E}\cap\mathcal{F}\cap\mathcal{G}] \geq 1-\epsilon$. Thus,

$$\mathbb{E}\left[\left(\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - r_t\right)_+\right] \leq 4\epsilon.$$

Because this holds for any $\epsilon > 0$, almost surely, $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - r_t \leq 0$. Thus, the learning rule is universally consistent on $\mathbb{X}$. This ends the proof. ■

To the best of our knowledge, while we believe that for general spaces $\mathcal{X}$ with non-atomic probability measures, one may have a gap $\mathcal{C}_5 \subsetneq \mathcal{C}_6$, it seems plausible that $\mathcal{C}_5 = \mathcal{C}_7$. As a consequence, this would imply that we have an exact characterization for processes admitting universal learning with prescient rewards $\textsc{Soab}_{prescient} = \mathcal{C}_5 = \mathcal{C}_7$.

Comparison to a more natural condition $\mathcal{C}_8$. In the rest of this section, we compare condition $\mathcal{C}_5$ to another potentially more natural sufficient condition. In Proposition 5.1 from Chapter 5, we showed that given any $\mathbb{X} \in$ SMV process, only allowing for a finite number of duplicates in $\mathbb{X}$ yields an extended CS process. Precisely, for any M, letting

$$\mathcal{T}^{\leq M} = \left\{ t \geq 1 : \sum_{t' \leq t} \mathbb{1}[X_{t'} = X_t] \leq M \right\},$$

be the set of times corresponding to duplicates having index at most M, we have that $\mathbb{X}^{\mathcal{T}^{\leq M}} \in$ CS. However, if one does not restrict the maximum number of duplicates, one loses the extended CS property. A natural condition on stochastic processes would therefore be that for some increasing rate of maximum number of duplicates, the CS property is conserved. For any process $\mathbb{X}$, we denote the occurrence count as $N_t(x) = \sum_{i=1}^{t} \mathbb{1}[X_t = x]$ for all $x \in \mathcal{X}$. Then, the condition on stochastic processes can be formally defined as follows.

Condition 8. *There exists an increasing function $\Psi : \mathbb{N} \to \mathbb{N}$ with $\Psi(T) \to \infty$ as $T \to \infty$ such that for any sequence of measurable sets $A_i \in \mathcal{B}$ for $i \geq 1$ with $A_i \downarrow \emptyset$,*

$$\mathbb{E}\left[\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{A_i}(X_t) \mathbb{1}_{N_t(X_t) \leq \Psi(T)} \right] \to 0.$$

Although this condition is indeed sufficient for universal learning, we show that the more involved $\mathcal{C}_5$ class of processes is larger, and strictly larger whenever $\mathcal{X}$ admits a non-atomic probability measure.

Proposition 6.7. *Let $\mathcal{X}$ be a metrizable separable Borel space, then $\mathcal{C}_8 \subset \mathcal{C}_5$. Further, if there exists a non-atomic probability measure on $\mathcal{X}$, then $\mathcal{C}_8 \subsetneq \mathcal{C}_5$.*

Proof We first show $\mathcal{C}_8 \subset \mathcal{C}_5$. Indeed, suppose that $\mathbb{X} \in \mathcal{C}_8$, then there exists $\Psi : \mathbb{N} \to \mathbb{N}$ increasing to infinity such that for any measurable sets $A_k \downarrow \emptyset$, we have

$$\mathbb{E}\left[\limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T, N_t(X_t) \leq \Psi(T)} \mathbb{1}_{A_k}(X_t) \right] \underset{k \to \infty}{\longrightarrow} 0.$$

Now let $T_i \geq 1$ such that $\Psi(T_i) \geq 1 + i2^i$. We now show that $(T_i)_i$ satisfies the condition of condition $\mathcal{C}_5$. Let $\mathcal{T} = \bigcup_{i \geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$, and $A_k \downarrow \emptyset$. For any $T \geq 1$, we denote $\mathcal{X}(T) = \{X_t, t \leq T\}$ the set of visited instances. Now fix $k \geq 0$. Then, for $T \geq T_k$, let $i \geq k$ such that $T_i \leq T < T_{i+1}$,

$$\begin{aligned} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}} \mathbb{1}_{A_k}(X_t) &\leq \frac{1}{2^k} + \frac{1}{T} \sum_{2^{-k}T < t \leq T, t \in \mathcal{T}} \mathbb{1}_{A_k}(X_t) \\ &= \frac{1}{2^k} + \frac{1}{T} \sum_{x \in \mathcal{X}(T) \cap A_k} |\{2^{-k}T < t \leq T, t \in \mathcal{T} : X_t = x\}|. \end{aligned}$$

In $\mathcal{T}$, we accept at most one duplicate per phase. Because $T_i \leq T < T_{i+1}$, the interval $[2^{-k}T, T]$ intersects at most $1 + k2^i$ phases. Thus, for any $x \in \mathcal{X}(T)$, $|\{2^{-k}T < t \leq T, t \in \mathcal{T} : X_t = x\}| \leq 1 + k2^i \leq 1 + i2^i \leq \Psi(T)$. Thus, for any $T \geq T_k$,

$$\begin{aligned}\frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_k}(X_t) &\leq \frac{1}{2^k} + \frac{1}{T}\sum_{x\in\mathcal{X}(T)\cap A_k} \min(|\{t \leq T : X_t = x\}|, \Psi(T)) \\ &= \frac{1}{2^k} + \frac{1}{T}\sum_{t\leq T, N_t(X_t)\leq\Psi(T)} \mathbb{1}_{A_k}(X_t).\end{aligned}$$

Using the hypothesis on Ψ applied to $A_k \downarrow \emptyset$ yields

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{A_k}(X_t)\right] \underset{k\to\infty}{\longrightarrow} 0.$$

Hence, this shows that $\mathbb{X}^{\mathcal{T}} \in \text{CS}$ and $\mathbb{X} \in \mathcal{C}_5$.

Next, suppose that there exists a non-atomic probability measure on $\mathcal{X}$. We will construct explicitly a process $\mathbb{X} \in \mathcal{C}_5 \setminus \mathcal{C}_8$. By Lemma 6.2, there exists a sequence of disjoint measurable sets $(A_i)_{i\geq 0}$ together with non-atomic probability measures $(\nu_i)_{i\geq 0}$ such that $\nu_i(A_i) = 1$. We now fix $x_0 \in A_0$ an arbitrary instance (we will not use the set A_0 any further) and define subsets of indices as follows, $S_i = \{k \geq 1 : k \equiv 2^{i-1} \bmod 2^i\}$. Note that the sets $(S_i)_{i\geq 1}$ form a partition of $\mathbb{N}$. We now introduce independent processes $\mathbb{Z}^i$ for $i \geq 1$ such that $\mathbb{Z}^i = (Z^i_t)_{t\geq 1}$ is an i.i.d. process with distribution ν_i. Last, for all $i \geq 1$ we denote $n_i = 2^{\lfloor \log_2 i \rfloor}$. Now consider the following process $\mathbb{X}$ where $X_1 = x_0$ and for any $t \geq 1$,

$$X_t = Z^i_{\lfloor \frac{t}{n_i} \rfloor}, \quad 2^k \leq t < 2^{k+1}, k \equiv 2^{i-1} \bmod 2^i.$$

When the process is in phase i, it corresponds to an i.i.d. process on A_i which is duplicated n_i times. Note that we used n_i duplicates instead of i so that each point is duplicated exactly n_i times (we do not have boundary issues at the end of the phase). We now show that $\mathbb{X} \notin \mathcal{C}_8$. Let $\Psi : \mathbb{N} \to \mathbb{N}$ an increasing function with $\Psi(T) \to \infty$ as $T \to \infty$. For $i \geq 1$, we first construct an increasing sequence of times T_i such that $\Psi(T_i) > n_i$. Then, for any $k \geq 1$, consider times $T^k = k2^i + 2^{i-1}$ which belong to S_i. Then, consider the event $\mathcal{F}_i$ such that the process $\mathbb{Z}^i$ only takes distinct values in A_i. Note that $\mathbb{P}[\mathcal{F}_i] = 1$ because the ν_i is non-atomic and $\nu_i(A_i) = 1$. Then, on $\mathcal{F}_i$, by construction, we have for any $k \geq 0$, with $T^k \geq T_i$,

$$\begin{aligned}\frac{1}{2T^k - 1}\sum_{t=1}^{2T^k-1} \mathbb{1}_{A_i}(X_t)\mathbb{1}_{N_t(X_t)\leq\Psi(2T^k-1)} &\geq \frac{1}{2T^k}\sum_{t=T^k}^{2T^k-1} \mathbb{1}_{A_i}(X_t)\mathbb{1}_{N_t(X_t)\leq n_i} \\ &= \frac{1}{2T^k}\sum_{t=T^k}^{2T^k-1} \mathbb{1}_{A_i}(X_t) \\ &\geq \frac{T^k}{2T^k - 1}.\end{aligned}$$

Hence, on the event $\mathcal{F}_i$, we have $\limsup_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{A_i}(X_t)\mathbb{1}_{N_t(X_t)\leq\Psi(T)} \geq \frac{1}{2}$. Because $\mathbb{P}[\mathcal{F}_i] = 1$, we obtain

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{A_i}(X_t)\mathbb{1}_{N_t(X_t)\leq\Psi(T)}\right] \geq \frac{1}{2}.$$

Now consider $B_i = \bigcup_{j\geq i} A_i$. Then, we have $B_i \downarrow \emptyset$ and for any $i \geq 1$,

$$\mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{B_i}(X_t)\mathbb{1}_{N_t(X_t)\leq\Psi(T)}\right] \geq \mathbb{E}\left[\limsup_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{A_i}(X_t)\mathbb{1}_{N_t(X_t)\leq\Psi(T)}\right] \geq \frac{1}{2}.$$

As a result, $\mathbb{X} \notin \mathcal{C}_8$.

We now show that $\mathbb{X} \in \mathcal{C}_5$. To do so, we first prove that $\mathbb{X} \in$ SMV. Let $(B_l)_{l\geq 1}$ be a sequence of disjoint measurable sets. Because $\mathbb{Z}^i$ are i.i.d. processes, we have $\mathbb{Z}^i \in$ SMV. In particular, on an event $\mathcal{E}_i$ of probability one, we have

$$|\{l : \mathbb{Z}^i_{\leq T} \cap B_l \neq \emptyset\}| = o(T).$$

Now consider the event $\mathcal{E} = \bigcap_{i\geq 1} \mathcal{E}_i$. This has probability one by the union bound. Let $\epsilon > 0$ and $i^* = \lceil \frac{2}{\epsilon} \rceil$. In particular, we have $\frac{1}{n_{i^*}} \leq \epsilon$. On the event $\mathcal{E}$, for any $i \leq i^*$, there exist T_i such that for all $T \geq T_i$,

$$|\{l : \mathbb{Z}^i_{\leq T} \cap B_l \neq \emptyset\}| \leq \frac{\epsilon}{2^i} T.$$

Now consider $T^0 = \max_{i\leq i^*} T_i n_i$. Then, for any $T \geq T^0$, we have

$$\begin{aligned}
|\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}| &\leq \sum_{i=1}^{i^*} |\{l : \mathbb{Z}^i_{\leq \lfloor T/n_i \rfloor} \cap B_l \neq \emptyset\}| \\
&\quad + |\{l : \exists t \leq T : X_t \in B_l, 2^k \leq t < 2^{k+1}, k \equiv 0 \bmod 2^{i^*}\}| \\
&\leq \epsilon T + |\{X_t, \quad t \leq T, 2^k \leq t < 2^{k+1}, k \equiv 0 \bmod 2^{i^*}\}| \\
&\leq \epsilon T + 2\frac{T}{n_{i^*}},
\end{aligned}$$

where in the last inequality we used the fact that in a phase $i > i^*$, each point is duplicated $n_i \geq n_{i^*}$ times. As a result, on the event $\mathcal{E}$, we have

$$\limsup \frac{|\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}|}{T} \leq 3\epsilon.$$

Because this holds for all $\epsilon > 0$, we obtain that on $\mathcal{E}$, $|\{l : \mathbb{X}_{\leq T} \cap B_l \neq \emptyset\}| = o(T)$. Because $\mathcal{E}$ has probability one, this ends the proof that $\mathbb{X} \in$ SMV. Now consider the following times $T_j = 4^j$ for $j \geq 0$ and define $\mathcal{T} = \bigcup_{j\geq 0} \mathcal{T}^j \cap \{t \geq T_i\}$. We aim to show $\mathbb{X}^{\mathcal{T}} \in$ CS. First, note that for any $j \geq 0$, the phases $[2^k, 2^{k+1})$ contained in $[T_j, T_{j+1})$ satisfy $k \leq 2j + 1$. Let $i(j) = 1 + \log_2(2j + 1)$. We have $k \in \bigcup_{i\leq i(j)} S_i$, which implies that each instance X_t is duplicated consecutively at most $n_{i(j)}$ times in $\mathbb{X}$ within $[T_j, t_{j+1})$. However, the sections defined by $\mathcal{T}$ have length at least $2^{-j}T_j = 2^j$. Further, all the phases were constructed so

that there are no boundary issues: if $n_{i(j)} \leq 2^j$, then $\mathcal{T}$ does not contain any duplicates during the period $[T_j, T_{j+1})$. Because $n_{i(j)} \leq i(j) = o(2^j)$, there exists $j_0 \geq 0$ such that $\mathcal{T}$ does not contain any duplicate on $[T_{j_0}, \infty)$. Let $\mathcal{T}(0) = \{t \geq 1 : N_t(X_t) = 1\}$ the set of first appearances. Then, for any $A \in \mathcal{B}$ and $T \geq 1$,

$$\sum_{t \leq T, t \in \mathcal{T}} \mathbb{1}_A(X_t) \leq T_{j_0} + \sum_{t \leq T, t \in \mathcal{T}(0)} \mathbb{1}_A(X_t).$$

Now because $\mathbb{X} \in \text{SMV}$, we have $\mathbb{X}^{\mathcal{T}(0)} \in \text{CS}$ which implies $\mathbb{X}^{\mathcal{T}} \in \text{CS}$ by the above inequality. This ends the proof of the proposition. ■

6.5.3 Universal learning with fixed excess error tolerance

In this section, we show that as an application of the methods developed in Chapter 5 and in this chapter, achieving a fixed excess regret $\epsilon > 0$ is always possible for SMV processes. This is stated in Proposition 6.1. We first need to recall a result from Chapter 5 showing that SMV processes without duplicates are CS extended processes.

Proposition 5.1. *Let $\mathbb{X}$ be a stochastic process on $\mathcal{X}$, and for $M \geq 1$, define an $\mathbb{X}$-dependent set*

$$\mathcal{T}^{\leq M} = \left\{ t \geq 1 : \sum_{t' \leq t} \mathbb{1}[X_{t'} = X_t] \leq M \right\},$$

the set of times which are duplicates with index at most M. In particular, $\mathcal{T}^{\leq 1}$ is the set of all times of first appearances *of values. Similarly, $\mathcal{T}^{<M} = \mathcal{T}^{\leq M-1}$ for $M \geq 2$. For brevity, we introduce the shorthand notation $\mathbb{X}^{(\leq M)} = \mathbb{X}^{\mathcal{T}^{\leq M}}$. The following are equivalent.*

1. *$\mathbb{X} \in \text{SMV}$.*
2. *$\mathbb{X}^{(\leq 1)} \in \text{CS}$.*
3. *For all $M \geq 1$, $\mathbb{X}^{(\leq M)} \in \text{CS}$.*

We are now ready to prove Proposition 6.1.

Proof of Proposition 6.1 We first describe the algorithm that depends on a parameter $M \geq 1$ which we will fix later. We use the notation $\mathcal{T}^{\leq M}$ from Proposition 5.1 for the set of times that are duplicates having index at most M. Note that whether $t \in \mathcal{T}^M$ or $t \notin \mathcal{T}^M$ can be decided in an online manner. Next we fix a sequence $\Pi = (\pi^l)_{l \geq 1}$ of policies that are dense within extended CS processes from Lemma 6.4. The learning rule f. simply performs the EXPINF strategy on the sequence Π for times in $\mathcal{T}^{\leq M}$ and for other times performs independent copies of the EXP3.IX algorithm in parallel for each distinct instance. Formally, for any $t \geq 1$, instances $\boldsymbol{x}_{\leq t}$ and observed rewards $\boldsymbol{r}_{\leq t-1}$, we define

$$f_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}, x_t) = \begin{cases} \text{EXPINF}(\boldsymbol{x}_{U_t}, \hat{\boldsymbol{a}}_{U_t}, \boldsymbol{r}_{U_t}, x_t) & \text{if } t \in \mathcal{T}^M \\ \text{EXP3.IX}_{\mathcal{A}}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t}) & \text{o.w.} \end{cases}$$

where $U_t = \{t' \leq t-1 : t \in \mathcal{T}^M\}$ and $S_t = \{t' < t : x_t = x_{t'}, t' \in \mathcal{T}^M\}$ and $\hat{a}_{t'}$ denotes the action selected at time $t' \leq t-1$.

Let $\mathbb{X} \in \text{SMV}$. We now prove that this learning rule achieves low excess error compared to a fixed measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$. We denote by $\hat{a}_t(M)$ its selected action at time t. First, by Proposition 5.1, $\mathbb{X}^{\mathcal{T}^M} \in \text{CS}$. Further, as discussed in Section 6.5.2, the same proof of universal consistency of EXPINF under CS processes for stationary rewards given in Chapter 5 shows that EXPINF is universally consistent under CS extended processes for adversarial rewards. This is a consequence of the fact that the regret guarantee of EXP3.IX (Theorem 5.6) holds for adversarial rewards as well. Thus, on an event $\mathcal{E}$ of probability one,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^M} r_t(\pi^*(X_t)) - r_t(\hat{a}_t(M)) \leq 0.$$

Next, similarly to the proof of Proposition 6.3, let $\epsilon(T) = \frac{1}{T}|\{X_t : t \leq T, t \notin \mathcal{T}^M\}|$. The same proof as in Proposition 6.3 shows that on an event $\mathcal{F}$ of probability one, for all $T \geq 1$,

$$\frac{1}{T} \sum_{t\leq T, t\notin\mathcal{T}^M} r_t(\pi^*(X_t)) - r_t(\hat{a}_t(M)) \\ \leq 8|\mathcal{A}|\frac{\ln T}{T^{1/4}} + 3c\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\frac{1}{\ln T} + \sqrt{\epsilon(T)} + 3\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}\epsilon(T)^{1/4}.$$

Note that to each element of $\{X_t : t \leq T, t \notin \mathcal{T}^M\}$ correspond least M duplicates in $\mathcal{T}^M$ so that $\epsilon(T) \leq \frac{1}{M}$. As a result, combining the two previous equations yields on the event $\mathcal{E} \cap \mathcal{F}$ of probability one,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^M} r_t(\pi^*(X_t)) - r_t(\hat{a}_t(M)) \leq 4\frac{\sqrt{|\mathcal{A}|\ln|\mathcal{A}|}}{M^{1/4}}.$$

Thus, taking $M \geq 4^4|\mathcal{A}|^2\ln^2|\mathcal{A}|\epsilon^{-4}$ gives a learning rule with the desired ϵ excess error almost surely. This ends the proof of the proposition. ■

6.6 Model Extensions

6.6.1 Infinite action spaces

The previous sections focused on the case of finite action spaces. For infinite action spaces, we argue that as a direct consequence of the analysis of the stationary case in Chapter 5, one can obtain a characterization of learnable processes and the same optimistically universal learning rules.

For countably infinite action spaces, we showed in Chapter 5 that EXPINF performed with the countable sequence of dense policies given by Lemma 6.4 is universally consistent under CS processes with stationary rewards, and that CS is necessary. As discussed in Sections 6.5.2 and 6.5.3, the same arguments as in Chapter 5 show that EXPINF is universally consistent under CS processes for adversarial rewards as well. Further, since adversarial

rewards generalize stationary rewards, CS is still necessary for universal learning. Thus, $\text{SOAB}_{online} = \text{SOAB}_{prescient} = \text{SOAB}_{oblivious} = \text{SOAB}_{memoryless} = \text{SOAB}_{stat} = \text{CS}$ and EXPINF is optimistically universal in all reward settings.

For uncountable separable metrizable Borel action spaces $\mathcal{A}$, even for stationary rewards, universal learning is impossible as we showed in Chapter 5. Hence, $\text{SOAB}_{online} = \text{SOAB}_{prescient} = \text{SOAB}_{oblivious} = \text{SOAB}_{memoryless} = \text{SOAB}_{stat} = \emptyset$.

6.6.2 Unbounded rewards

We now turn to the case of unbounded rewards $\mathcal{R} = [0, \infty)$. We further suppose that for any $t \geq 1$, and history $\boldsymbol{x} \in \mathcal{X}^\infty, \boldsymbol{a}_{\leq t} \in \mathcal{A}^t, \boldsymbol{r}_{\leq t-1} \in \mathcal{R}^{t-1}$, the random variable $r_t(a_t \mid \mathbb{X} = \boldsymbol{x}, \hat{\boldsymbol{a}}_{\leq \boldsymbol{t-1}} = \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}(\hat{a})_{\leq t-1} = \boldsymbol{r}_{\leq t-1})$ is integrable so that the immediate expected reward is well defined. Again, in this case, adversarial rewards yield the same results as stationary rewards. Clearly, for uncountable separable metrizable Borel action spaces, under unbounded rewards, universal learning is still impossible $\text{SOAB}^{unbounded}_{online} = \text{SOAB}^{unbounded}_{prescient} = \text{SOAB}^{unbounded}_{oblivious} = \text{SOAB}^{unbounded}_{memoryless} = \text{SOAB}^{unbounded}_{stat} = \emptyset$, because this was already the case for bounded rewards.

For countable action spaces $\mathcal{A}$, condition FS is necessary even under the full-feedback noiseless setting (Chapter 3), hence necessary for contextual bandits as well. Also, in Chapter 5, we proposed the algorithm that runs an independent EXPINF learner on each distinct context instance, which is universally consistent under FS processes. As in the previous section, this guarantee still holds for adversarial rewards, and FS is still necessary for universal learning. Therefore, $\text{SOAB}^{unbounded}_{online} = \text{SOAB}^{unbounded}_{prescient} = \text{SOAB}^{unbounded}_{oblivious} = \text{SOAB}^{unbounded}_{memoryless} = \text{SOAB}^{unbounded}_{stat} = \text{FS}$.

6.6.3 Uniformly-continuous rewards

We assume that the rewards are bounded again. In the previous sections, we showed that for finite action sets, universal learning is possibly under large classes of processes, namely at least on $\mathcal{C}_5$ processes. However, for countable action sets, this is reduced to CS, and for uncountable action sets, universal learning is not achievable. Therefore, imposing no constraints on the rewards is too restrictive for universal learning in the last cases. Here, we investigate the case when $\mathcal{A}$ is a separable metric space given with a metric d, and the rewards are uniformly-continuous. Crucially, the modulus of continuity should be uniformly bounded over time as well. We recall the definition of uniformly-continuous rewards.

Let $(\mathcal{A}, d)$ be a separable metric space. The reward mechanism $(r_t)_{t\geq 1}$ is uniformly-continuous if for any $\epsilon > 0$, there exists $\Delta(\epsilon) > 0$ such that

$$\forall t \geq 1, \forall (\boldsymbol{x}_{\leq t}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}) \in \mathcal{X}^t \times \mathcal{A}^{t-1} \times \mathcal{R}^{t-1}, \forall a, a' \in \mathcal{A},$$
$$d(a, a') \leq \Delta(\epsilon) \Rightarrow |\mathbb{E}[r_t(a) - r_t(a') \mid \mathbb{X}_{\leq t} = \boldsymbol{x}_{\leq t}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}]| \leq \epsilon,$$

In the definition, the expectation is taken over the rewards' randomness, in the event when the context sequence until t is exactly $\boldsymbol{x}_{\leq t}$, the learner selected actions $\boldsymbol{a}_{\leq t-1}$ and received rewards $\boldsymbol{r}_{\leq t-1}$ in the first $t-1$ steps. For instance, for stationary rewards, only x_t is relevant in this expectation, while for online rewards, $\boldsymbol{x}_{\leq t}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}$ may be relevant.

The above definition is not written for prescient rewards for simplicity. For these, we need to condition on the complete sequence $\mathbb{X}$:

$$\forall t \geq 1, \forall (\boldsymbol{x}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}) \in \mathcal{X}^\infty \times \mathcal{A}^{t-1} \times \mathcal{R}^{t-1}, \forall a, a' \in \mathcal{A},$$
$$d(a, a') \leq \Delta(\epsilon) \Rightarrow |\mathbb{E}[r_t(a) - r_t(a') \mid \mathbb{X} = \boldsymbol{x}, \boldsymbol{a}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}]| \leq \epsilon.$$

As in the case of unrestricted rewards, we consider the set of processes $\textsc{Soab-Uc}_{setting}$ admitting universal learning for uniformly-continuous rewards under any chosen reward setting. The uniform-continuity assumption defined above generalizes the corresponding assumption proposed in Chapter 5 for stationary rewards. There, we also proposed a weaker continuity assumption on the immediate expected rewards, however, similarly as in Section 6.6.1 one can easily check that with this reward assumption, adversarial settings give the same results as the stationary case.

The goal of this section is to show that under the mild uniform-continuity assumption on the rewards, when the action space is totally-bounded, one can recover all the results from the finite action space case. We first start by showing that the derived necessary conditions still hold. To do so, we will use the following reduction lemma.

Lemma 6.5. *Let $\mathcal{X}$ be a metrizable separable Borel space and let $(\mathcal{A}, d)$ be a separable metric space. Let $S \subset \mathcal{A}$ such that $\min_{a,a' \in S} d(a, a') > 0$. Then, we have $\textsc{Soab-Uc}_{setting}(\mathcal{A}) \subset \textsc{Soab}_{setting}(S)$ for any setting $\in \{stat, memoryless, oblivious, prescient, online\}$.*

Further, if there is a learning rule for uniformly continuous rewards in $\mathcal{A}$ that is universally consistent under a set of processes $\tilde{\mathcal{C}}$ on $\mathcal{X}$, there is also a learning rule for unrestricted rewards in S that is universally consistent under all $\tilde{\mathcal{C}}$ processes.

Proof The first claim was proven in Chapter 5 for the specific case of stationary rewards. There, we showed that the case of uniformly-continuous rewards on $\mathcal{A}$ is at least harder than the unrestricted rewards on S through a simple reduction. Here, we show that the reduction can be extended to adversarial rewards as well. Denote $\eta = \frac{1}{3} \min_{a,a' \in S} d(a, a')$. Any realization $r : S \to [0, 1]$ can be extended to a $1/\eta$-Lipschitz function $F(r) : \mathcal{X} \to \mathcal{A}$ by

$$F(r)(a) = \max\left(0, \max_{a' \in S} r(a') - d(a, a') \frac{\bar{r}}{\eta}\right), \quad a \in \mathcal{A}.$$

Then, a general reward mechanism $(r_t)_{t \geq 1}$ on S can be extended to a reward mechanism on $\mathcal{A}$ such that for any realization, $r_t : a \in \mathcal{A} \to [0, 1]$ is $1/\eta$-Lipschitz. Hence, the mechanism $(r_t)_{t \geq 1}$ is uniformly-continuous. From now, the same arguments as in the proof of Lemma 5.5 from Chapter 5 show that the reduction holds and that $\textsc{Soab-Uc}_{setting}(\mathcal{A}) \subset \textsc{Soab}_{setting}$ for the considered setting. Intuitively, since for any realization, $r_t : a \in \mathcal{A} \to [0, 1]$ has zero value outside of the balls $B_d(a, \eta)$ for $a \in S$, that on the ball $B_d(a, \eta)$ for $a \in S$, the action a has maximum reward, and that these balls are disjoint, without loss of generality, one can assume that a universally consistent learning rule always selects actions in S under these rewards, in which case, the problem becomes equivalent to having unrestricted rewards on the action set S. The formal learning rule reduction is defined in the original proof, and one can check that the reduction is invariant in the process $\mathbb{X}$. Hence, this also proves the second claim of the lemma. ■

This lemma allows to use the necessary conditions for the unrestricted reward setting by changing the terms "finite action set" (resp. "countably infinite action set") into "totally-bounded action set" (resp. "non-totally-bounded action set"). The second claim of Lemma 6.5 will be useful to show that no optimistically universal learning exists for adversarial uniformly-continuous rewards either. More precisely, the following result is a direct consequence of the first claim of Lemma 6.5.

Proposition 6.8. *Let $\mathcal{X}$ be a metrizable separable Borel space and let $\mathcal{A}$ be a non-totally-bounded metric space. Then, for any reward setting, SOAB-UC $\subset$ CS.*

Let $\mathcal{A}$ be a totally-bounded metric space with $|\mathcal{A}| > 2$. Then, for any reward setting, SOAB-UC $\subset$ SMV. Further, if $\mathcal{X}$ admits a non-atomic probability measure, we have that SOAB-UC$_{memoryless} \subsetneq$ SMV, SOAB-UC$oblivious \subset \mathcal{C}_6$ and SOAB-UC$_{prescient} \subset \mathcal{C}_7$.

We now show that we can recover the sufficient conditions from previous sections as well. For uniformly-continuous rewards, we can show that there exists a countable set of dense policies under extended CS processes, as was the case for unrestricted rewards and countable action sets.

Lemma 6.6. *Let $\mathcal{A}$ be a separable metric space. There is a countable set of measurable policies Π such that for any extended process $\mathbb{X}^{\mathcal{T}} \in$ CS, any measurable policy $\pi^* : \mathcal{X} \to \mathcal{A}$, and any uniformly-continuous possibly stochastic rewards $(r_t)_t$, with probability one over the rewards,*

$$\begin{cases} \inf_{\pi\in\Pi} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} r_t(\pi^*(X_t)) - r_t(\pi(X_t)) \leq 0, \\ \inf_{\pi\in\Pi} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi(X_t)) \leq 0, \end{cases}$$

where $\bar{r}_t = \mathbb{E}r_t$ is the immediate average reward.

Proof For any $\epsilon > 0$, let $\Delta(\epsilon)$ be the $\epsilon-$modulus of continuity of the sequence of rewards $(\bar{r}_t)_t$. By Lemma 5.4 from Chapter 5 (and with a straightforward adaptation for extended processes), on an event $\mathcal{E}$ of probability one, for any $i \geq 1$, there exists $\pi^i \in \Pi$ such that $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}[d(\pi^*(X_t), \pi^i(X_t)) \geq 2^{-i}] \leq 2^{-i}$, and for all $i \geq 1$, we have that $\frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} r_t(\pi^i(X_t)) - \bar{r}_t(\pi^i(X_t)) \to 0$ and similarly for π^*, where $\bar{r}_t$ is the immediate expected reward at time t. We now suppose that this event is met. Let $\epsilon > 0$, let $i \geq 1$ such that $2^{-i} \leq \Delta(\epsilon)$. Then,

$$\begin{aligned} \sum_{t\leq T, t\in\mathcal{T}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^i(X_t)) &\leq \sum_{t\leq T, t\in\mathcal{T}} (\bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^i(X_t)))\mathbb{1}_{d(\pi^i(x),\pi^*(x))<\Delta(\epsilon)} \\ &\quad + \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{d(\pi(x),\pi^*(x))\geq 2^{-i}} \\ &\leq \epsilon T + \sum_{t\leq T, t\in\mathcal{T}} \mathbb{1}_{d(\pi(x),\pi^*(x))\geq 2^{-i}}. \end{aligned}$$

As a result, $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^i(X_t)) \leq \epsilon + \Delta(\epsilon)$. Further, because the event $\mathcal{E}$ is satisfied, $\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} r_t(\pi^*(X_t)) - r_t(\pi^i(X_t)) \leq \epsilon + \Delta(\epsilon)$. This holds for any $\epsilon > 0$. Now because $\Delta(\epsilon) \to 0$ as $\epsilon \to 0$, we proved that on $\mathcal{E}$,

$$\begin{cases} \inf_{\pi\in\Pi} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} r_t(\pi^*(X_t)) - r_t(\pi(X_t)) \leq 0, \\ \inf_{\pi\in\Pi} \limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi(X_t)) \leq 0. \end{cases}$$

This ends the proof of the lemma. ■

We are now ready to generalize our algorithms from previous sections, using Π as a countable set of functions that are dense within all policies in the uniformly-continuous rewards context. First, note that using EXPINF directly with the countable family described in Lemma 6.6 is universally consistent on all CS processes. This shows that we always have CS $\subset$ SOAB-UC for all models. In particular, together with Proposition 6.8, this shows that for non-totally-bounded metric action spaces $\mathcal{A}$, we have SOAB-UC = CS for all reward models.

Next, we turn to the case of finite action spaces and context spaces $\mathcal{X}$ that do not admit a non-atomic measure. In this case, we showed that the algorithm that simply uses different EXP3.IX for each distinct instance is optimistically universal. In the case of uniformly-continuous rewards, we can replace EXP3.IX with EXPINF over a countable set of actions. This yields an optimistically universal learning rule for any totally-bounded action spaces $\mathcal{A}$.

Theorem 6.13. *Let $\mathcal{X}$ be a metrizable separable Borel space that does not admit a non-atomic probability measure. Let $\mathcal{A}$ be a totally-bounded metric space. Then, there exists an optimistically universal learning rule for uniformly-continuous rewards (in any setting) and learnable processes are exactly SOAB-UC$_{stat}$ = SOAB-UC$_{online}$ = SMV.*

Proof We first describe the learning rule. For any $\epsilon > 0$, let $\mathcal{A}(\epsilon)$ be an $\epsilon-$net of $\mathcal{A}$. By abuse of notation, for any $a \in \mathcal{A}$, we use the same notation a for the expert which selects action a at all time steps. Now consider the countable set of experts $\bigcup_{i\geq 1} \mathcal{A}(2^{-i}) = \{a_1, a_2, \ldots\}$, where the sets are concatenated by increasing order of index i. Now consider the learning rule that uses a distinct EXPINF over this set of experts, for each distinct instance. Formally, the learning rule is

$$f_t(\boldsymbol{x}_{\leq t-1}, \boldsymbol{r}_{\leq t-1}, x_t) = \text{EXPINF}(\hat{\boldsymbol{a}}_{S_t}, \boldsymbol{r}_{S_t})$$

where $S_t = \{t' < t : x_{t'} = x_t\}$ is the set of times that x_t was visited previously and $\hat{a}_{t'}$ denotes the action selected at time t' for $t' < t$. We now show that this learning rule is universally consistent on all SMV processes for uniformly bounded rewards. In the proof of Theorem 6.5 we showed that for spaces $\mathcal{X}$ that do not admit a non-atomic probability measure, any SMV process visits a sublinear number of distinct instances almost surely. Therefore, for $\mathbb{X} \in$ SMV, on an event $\mathcal{E}$ of probability one, we have $|\{x \in \mathcal{X} : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}| = o(T)$. It now suffices to adapt the proof of Proposition 6.3. Let $(r_t)_t$ be a uniformly continuous reward mechanism. For $\epsilon > 0$, let $\Delta(\epsilon) > 0$ its $\epsilon-$modulus of continuity. We keep the same notations as in the proof of Proposition 6.3. Let $S_T = \{x : \{x\} \cap \mathbb{X}_{\leq T} \neq \emptyset\}$, $\epsilon(T) = \frac{|S_T|}{T}$ and for $x \in S_T$, let $\mathcal{T}_T(x) = \{t \leq T : X_t = x\}$. Further, for any $x \in S_T$ we pose $\mathcal{T}_T(x) = \{t \leq T : X_t = x\}$. Let $\mathcal{H}_0(T) = \{x \in S_T : |\mathcal{T}_T(x)| < \frac{1}{\sqrt{\epsilon(T)}}\}$, $\mathcal{H}_1(T) = \{x \in S_T : \frac{1}{\sqrt{\epsilon(T)}} \leq |\mathcal{T}_T(x)| < \ln^8 T\}$ and $\mathcal{H}_2(T) = \{x \in S_T : |\mathcal{T}_T(x)| \geq \ln^8 T\}$. Now let $\pi : \mathcal{X} \to \mathcal{A}$ be a measurable policy. We still have

$$\frac{1}{T} \sum_{x \in \mathcal{H}_0(T)} |\mathcal{T}_T(x)| \leq \sqrt{\epsilon(T)}.$$

Next, we turn to points $x \in \mathcal{H}_2(T)$. By Corollary 5.1, conditionally on the realization $\mathbb{X}$, for

any $x \in \mathcal{H}_2(T)$, with probability at least $1 - \frac{1}{T^3}$,

$$\max_{i \le \ln T} \sum_{t \in \mathcal{T}_T(x)} r_t(a_i) - r_t(\hat{a}_t) \le 4c|\mathcal{T}_T(x)|^{3/4}(\ln T)^{3/2} \le 4c\frac{|\mathcal{T}_T(x)|}{\sqrt{\ln T}}.$$

Therefore, since $|\mathcal{H}_2(T)| \le T$, by union bound, with probability at least $1 - \frac{1}{T^2} := 1 - p_2(T)$,

$$\sum_{x \in \mathcal{H}_2(T)} \max_{i \le \ln T} \sum_{t \in \mathcal{T}_T(x)} r_t(a_i) - r_t(\hat{a}_t) \le 4c\frac{T}{\sqrt{\ln T}}.$$

We then treat points in $\mathcal{H}_1(T)$ for which we will need to go back to the proof of the regret bounds for EXPINF and the underlying EXP3.IX algorithm which is used as a subroutine. First, we recall the structure of EXPINF. Let $i(k) = \sum_{r<k} r^3$. It works by periods $[i(k) + 1, i(k) + k^3)$ on which a new EXP3.IX learner finds the best expert within the first k experts in the sequence provided to EXPINF. We will refer to this as period k. As useful inequalities, we have $\frac{k^4}{4} \le i(k) \le \frac{(k+1)^4}{4}$. Let $k_0 = \lceil \epsilon(T)^{-1/8} \rceil$ and focus on a period k for $k \ge k_0$ of an EXPINF run. We denote by $\hat{a}_u$ the action selected at horizon u by EXPINF. Following the same arguments as in Proposition 6.3 and the analysis of EXP3.IX in [Neu15], for any $j \le k_0$

$$\sum_{u=1}^{k^3} (\ell_{u,\hat{a}_{i(k)+u}} - \tilde{\ell}_{u,a_j}) \le \frac{\ln k}{\eta_{k^3}} + \sum_{u=1}^{k^3} \eta_u \sum_{i=1}^{k} \tilde{\ell}_{u,a_i}.$$

As a result,

$$\sum_{u=1}^{k^3} \ell_{u,\hat{a}_{i(k)+u}} - \ell_{u,a_j} \le 3\sqrt{k \ln k \cdot k^3} + \sum_{u=1}^{k^3} (\tilde{\ell}_{u,a_j} - \ell_{u,a_j}) + \sum_{u=1}^{k^3} \sum_{i=1}^{k} \eta_u (\tilde{\ell}_{u,a_j} - \ell_{u,a_j})$$

Now for any $a \in \mathcal{A}$, let $a^{(k_0)} = \arg\min_{1 \le i \le k_0} d(a, a_i)$ the nearest neighbor of a where ties are broken alphabetically. We will sum this inequality for all EXPINF runs for $x \in \mathcal{H}_1(T)$, and periods $k \ge k_0$ that were completed, i.e. $|\mathcal{T}_T(x)| \ge i(k+1)$, taking $a_j = \pi(x)^{(k_0)}$. Before doing so, note that $\sum_{k' \le k} \sqrt{3(k')^4 \ln k'} \le (k+1)^3 \sqrt{\ln k} \le 4i(k+1)^{3/4}\sqrt{\ln i(k+1)}$. Further, for simplicity, denote by $A(T)$ (resp. $B(T)$) the sum that is obtained after summing all the terms $\sum_{u=1}^{k^3} (\tilde{\ell}_{u,a_j} - \ell_{u,a_j})$ (resp. $\sum_{u=1}^{k^3} \sum_{i=1}^{k} \eta_u (\tilde{\ell}_{u,a_j} - \ell_{u,a_j})$). Using these notations, we obtain

$$\begin{aligned}
&\sum_{x \in \mathcal{H}_1(T)} \sum_{t \in \mathcal{T}_T(x)} r_t(\pi(X_t)^{(k_0)}) - r_t(\hat{a}_t) \\
&\qquad \le \sum_{x \in \mathcal{H}_1(T)} \left(\frac{k_0^4}{4} + 4|\mathcal{T}_T(x)|^{3/4} + 4|\mathcal{T}_T(x)|^{3/4}\sqrt{3 \ln |\mathcal{T}_T(x)|} \right) + A(T) + B(T).
\end{aligned}$$

where in the first inequality, $\frac{k_0^4}{4}$ accounts for the first k_0 initial periods and $4|\mathcal{T}_T(x)|^{3/4}$ accounts for the last phase which potentially was not completed. Now recall that for each $x \in \mathcal{H}_1(T)$, $\epsilon(T)^{-1/2} \le |\mathcal{T}_T(x)| < \ln^8 T$. Let $n_0 \ge 1$ such that for any $n \ge n_0$, $8n^{3/4}\sqrt{3 \ln n} \le$

$n^{7/8}$. Since on the event $\mathcal{E}$, we have $\epsilon(T) \to 0$, there exists an index $\hat{T}$ such that for $T \geq \hat{T}$, $\epsilon(T)^{-1/2} \geq n_0$. Therefore, on $\mathcal{E}$, for $T \geq \hat{T}$ we have

$$\sum_{x \in \mathcal{H}_1(T)} \left(\frac{k_0^4}{4} + 20|\mathcal{T}_T(x)|^{3/4} + |\mathcal{T}_T(x)|^3 \sqrt{3 \ln |\mathcal{T}_T(x)|} \right) \leq 2\sqrt{\epsilon(T)}T + \sum_{x \in \mathcal{H}_1(T)} |\mathcal{T}_T(x)|^{7/8}$$
$$\leq (2\sqrt{\epsilon(T)} + \epsilon(T)^{1/16})T.$$

Next, using the same arguments as in the proof of Proposition 6.3, observe that conditionally on $\mathbb{X}$, $(A(T'))_{T' \leq T}$ is a super-martingale, with increments bounded in absolute value by $2\sqrt{\frac{k \cdot k^3}{\ln k}} \leq 2k^2 \leq 4\sqrt{i(k+1)} \leq 4 \ln^4 T$. Therefore, Azuma's inequality implies that

$$\mathbb{P}[A(T) \leq 8T^{3/4} \ln^4 T \mid \mathbb{X}] \geq 1 - e^{-2\sqrt{T}}.$$

Similarly, $(B(T'))_{T' \leq T}$ is a super-martingale, with increments bounded in absolute value by $2k\sqrt{\frac{k \cdot k^3}{\ln k}} \leq 8i(k+1) \leq 8 \ln^8 T$. Therefore,

$$\mathbb{P}[B(T) \leq 16T^{3/4} \ln^8 T \mid \mathbb{X}] \geq 1 - e^{-2\sqrt{T}}.$$

Therefore, by the Borel-Cantelli lemma, on an event $\mathcal{G}$ of probability one, we have that $\limsup_{T \to \infty} \frac{1}{T}(A(T)+B(T)) \leq 0$. Finally, let $j(T) = \min(\epsilon(T)^{-1/8}, \ln T)$. Putting everything together, we proved that on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, for $T \geq \hat{T}$,

$$\frac{1}{T} \sum_{t \leq T} r_t(\pi(X_t)^{(j(T))}) - r_t(\hat{a}_t) \leq 3\sqrt{\epsilon(T)} + \epsilon(T)^{1/16} + \frac{4c}{\sqrt{\ln T}} + \frac{1}{T}(A(T) + B(T)).$$

In particular, this shows that on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T} r_t(\pi(X_t)^{(j(T))}) - r_t(\hat{a}_t) \leq 0.$$

Now using Hoeffding's bound, with probability at least $1 - 2e^{-2\sqrt{T}}$, we have

$$\left| \sum_{t=1}^{T} r_t(\pi(X_t)^{(j(T))}) - \bar{r}_t(\pi(X_t)^{(j(T))}) \right| \leq 2T^{3/4}.$$

We have the same bound for π. Therefore, the Borel-Cantelli lemma implies that on an event $\mathcal{H}$ of probability one, $\frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)^{(j(T))}) - \bar{r}_t(\pi(X_t)^{(j(T))}) \to 0$ and $\frac{1}{T}\sum_{t=1}^{T} r_t(\pi(X_t)) - \bar{r}_t(\pi(X_t)) \to 0$. We now suppose that $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} \cap \mathcal{H}$ is met.

Now fix $\epsilon > 0$. Let k_0 such that $2^{-k_0} \leq \Delta(\epsilon)$. Because $\mathcal{E}$ is met, $\epsilon(T) \to 0$ and $j(T) \to \infty$. Thus, there exists $\tilde{T} \geq \hat{T}$ such that for any $T \geq \tilde{T}$, $\epsilon(T) \leq n_0^{-2}$ and $\mathcal{A}(2^{-k_0}) \subset \{a_i, j \leq j(T)\}$. Now for $T \geq \tilde{T}$ and any $a \in \mathcal{A}$, we have $d(a, a^{(j(T))}) \leq \Delta(\epsilon)$. As a result, using $\mathcal{H}$,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi(X_t)) - r_t(\hat{a}_t) \leq \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \bar{r}_t(\pi(X_t)) - \bar{r}_t(\pi(X_t)^{(j(T))})$$
$$\leq \limsup_{T \to \infty} \frac{\tilde{T}}{T} + \epsilon$$
$$\leq \epsilon.$$

In the second inequality, we used the uniform-continuity assumption on the rewards and the fact that for $T \geq \tilde{T}$, $d(\pi(X_t), \pi(X_t)^{(j(T))}) \leq \min_{a \in \mathcal{A}(2^{-k_0})} d(a, \pi(X_t)) \leq 2^{-k_0} \leq \Delta(\epsilon)$. Because this holds for any $\epsilon > 0$ and $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} \cap \mathcal{H}$ has probability one, this proves that the learning rule is universally consistent under $\mathbb{X}$. Then, the learning rule is universally consistent under any SMV process. By Proposition 6.8, this shows that the learnable processes are exactly SMV and that this is an optimistically universal learning rule. This ends the proof of the theorem. ■

The last algorithms needed to be adapted to the uniformly-continuous rewards setting are the algorithms for $\mathcal{C}_5$ processes in finite action spaces. Precisely, we will show that for totally-bounded metric action spaces $\mathcal{A}$, the set of learnable processes for uniformly-continuous adversarial rewards contains $\mathcal{C}_5$ processes. Recall that the class of constructed algorithms in Theorem 6.12 proceed separately on different categories of times. The category of t is defined based on the number of duplicates of X_t within its associated period. For each category of times, the learning rule performs a form of Hedge algorithm to perform the best strategy among strategy 0 which simply assigns a different EXP3.IX learner to distinct instances from the period; and strategy j for $j \geq 1$ which selected actions according to a fixed policy π^j, where $\tilde{\Pi} = \{\pi^l, l \geq 1\}$ was a dense set of policies within extended CS processes.

We make the following modifications to these learning rules. First, we replace $\tilde{\Pi}$ with the countable set Π of measurable policies that are dense in the uniformly-continuous rewards setting, as given by Lemma 6.6. Second, for every category p, strategy 0 will use EXP3.IX learners from $\mathcal{A}(\gamma_p)$, a γ_p-nets of $\mathcal{A}$, where γ_p is to be defined. With these modifications, we obtain the following result.

Theorem 6.14. *Let $\mathcal{X}$ be a metrizable separable Borel space and let $\mathcal{A}$ be a totally-bounded metric space. Then, $\mathcal{C}_5 \subset$ SOAB-UC_{online}.*

Proof Fix $\mathbb{X} \in \mathcal{C}_5$ and let $(T_i)_{i \geq 0}$ such that with $\mathcal{T} = \bigcup_{i \geq 0} \mathcal{T}^i \cap \{t \geq T_i\}$, we have $\mathbb{X}^{\mathcal{T}} \in$ CS. We first define how we modify the learning rule from Theorem 6.12 for this process. The functions PHASE, STAGE, PERIOD, CATEGORY are left unchanged. In the initial phase when $t < 2^{u(16p)}$, we replace EXP3.IX$_{\mathcal{A}}$ with EXPINF run with the dense sequence of $\mathcal{A}$ with the specific order described in the previous Theorem 6.13. We briefly recap the procedure. Let $\mathcal{A}(\epsilon)$ be an $\epsilon-$net of $\mathcal{A}$. We consider the sequence of experts $\bigcup_{i \geq 1} \mathcal{A}(2^{-i})$ where we confuse $a \in \mathcal{A}$ with the constant policy equal to a and we concatenate the nets by increasing order of index i. EXPINF is then run with this sequence of experts. Next, we enumerate $\Pi = \{\pi^l, l \geq 1\}$ and use these policies as well for the learning rule (strategies $j \geq 1$). Last, when playing strategy 0 after the initial phase, we replace EXP3.IX$_{\mathcal{A}}$ with EXP3.IX$_{\mathcal{A}(\gamma_p)}$, where γ_p will be defined shortly. In the original proof, we defined $\delta_p := 6\frac{\sqrt{|\mathcal{A}| \ln |\mathcal{A}|}}{2^p}$, $\eta_i := \sqrt{\frac{8 \ln(i+1)}{2^i}}$ and showed that the average error of the learning rule on $\mathcal{T}_p$ outside of the initial phase is $\mathcal{O}(\delta_p + \frac{\eta_{16p}}{4})$. Then, $\sum_{p \geq 0}(\delta_p + \frac{\eta_{16p}}{4}) < \infty$ allowed the learner to converge separately on each $\mathcal{T}_p$. We now replace δ_p with $\delta_p := 4\sqrt{\frac{|\mathcal{A}(\gamma_p)| \ln |\mathcal{A}(\gamma_p)|}{2^p}}$ and choose γ_p such that $\sum_p \delta_p < \infty$. We pose

$$\gamma_p = \min\{2^{-i} : |\mathcal{A}(2^{-i})| \ln |\mathcal{A}(2^{-i})| \leq 2^{p/4}\}.$$

Thus, we still have $\sum_p \delta_p < \infty$ and $\gamma_p \to 0$. We now show that the modified learning rule is universally consistent under online uniformly-continuous rewards on $\mathcal{A}$. Fix $(r_t)_t$ such a

reward mechanism and for $\epsilon > 0$, let $\Delta(\epsilon)$ the $\epsilon-$modulus of continuity of the sequence of immediate rewards. As in the original proof of Theorem 6.12, let $\mathcal{T}^{init} = \bigcup_{p\geq 0}\{t \in \mathcal{T}_p : t < 2^{u(16p)}\}$ be the initial phase. The process $(X_t)_{t\in\mathcal{T}^{init}}$ still visits a sublinear number of distinct instances almost surely, where we say that two instances $t, t' \in \mathcal{T}^{init}$ are duplicates if and only if they have same category, period and $X_t = X_{t'}$. As a result, in the proof of Theorem 6.13, we showed that for any $\pi^* : \mathcal{X} \to \mathcal{A}$, on an event $\mathcal{E}$ of probability one,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t\leq T, t\in\mathcal{T}^{init}} r_t(\pi^*(X_t)) - r_t \leq 0.$$

We then turn to non-initial phases and adapt the original proof of Theorem 6.12. For any $a \in \mathcal{A}$, we denote $a^{(\gamma)} = \arg\min_{a'\in\mathcal{A}(\gamma)} d(a, a')$, the nearest neighbor of a within the $\gamma-$net where ties are broken alphabetically. Keeping the same event $\mathcal{F}$, Eq (6.16) is unchanged and Eq (6.17) becomes

$$R_p(l, k; 0) \geq \sum_{t\in\mathcal{T}_p(l,k)} r_t(\pi^*(X_t)^{(\gamma_p)}) - 16|\mathcal{A}(\gamma_p)|^2 2^{-i} 2^{15l/16} - \delta_p 2^{l-i}.$$

Eq (6.18) is left unchanged. For $p \geq 0$, let $\epsilon(p) = \min\{2^{-i} : \gamma_p \leq \Delta(2^{-i})\}$. Note that because $\gamma_p \to 0$, we have $\epsilon(p) \to 0$ as $p \to \infty$. Following the same arguments as in the original proof and noting that $|\mathcal{A}(\gamma_p)| \leq 2^{p/4}$, Eq (6.19) is replaced by

$$\begin{aligned}\sum_{2^{u(16p)}<t\leq T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)^{(\gamma_p)}) - r_t &\leq 2^{\hat{l}} + c2^{p/2}T^{15/16} + \left(\delta_p + \frac{\eta_{16p}}{4}\right) T \\ &\leq 2^{\hat{l}} + cT^{31/32} + \left(\delta_p + \frac{\eta_{16p}}{4}\right).\end{aligned}$$

Now fix $\epsilon > 0$, and let p_0 such that $\sum_{p\geq p_0}(\delta_p + \frac{\eta_{16p}}{4}) < \epsilon$ and $\epsilon(p_0) < \epsilon$. Following the original arguments,

$$\sum_{p\geq p_0} \sum_{2^{u(16p)}\leq t<T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)^{(\gamma_p)}) - r_t \leq 2^{\hat{l}} \log_4 T + cT^{31/32} \log_4 T + \epsilon T.$$

Now using Azuma's inequality, with probability at least $1 - 4e^{-2\sqrt{T}}$, we have

$$\left|\sum_{p\geq p_0} \sum_{2^{u(16p)}\leq t<T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)^{(\gamma_p)}) - \bar{r}_t(\pi^*(X_t)^{(\gamma_p)})\right| \leq 2T^{3/4}$$

$$\left|\sum_{p\geq p_0} \sum_{2^{u(16p)}\leq t<T, t\in\mathcal{T}_p} r_t(\pi^*(X_t)) - \bar{r}_t(\pi^*(X_t))\right| \leq 2T^{3/4}.$$

Therefore, using Borel-Cantelli, on an event $\mathcal{G}$ of probability one, there exists $\hat{T}_1$ such that

for $T \geq \hat{T}_1$, the above two equations hold. Then, on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, for T sufficiently large,

$$\begin{aligned}\sum_{p \geq p_0} \sum_{2^{u(16p)} \leq t < T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t &\leq 2^{\hat{l}} \log_4 T + cT^{31/32} \log_4 T + \epsilon T + 4T^{3/4} \\ &\quad + \sum_{p \geq p_0} \sum_{2^{u(16p)} \leq t < T, t \in \mathcal{T}_p} \bar{r}_t(\pi^*(X_t)) - \bar{r}_t(\pi^*(X_t)^{(\gamma_p)}) \\ &\leq 2^{\hat{l}} \log_4 T + 4T^{3/4} + cT^{31/32} \log_4 T + 2\epsilon T,\end{aligned}$$

where in the last inequality we used the uniform continuity of the immediate expected rewards since for $p \geq p_0$, one has $\gamma_p \leq \gamma_{p_0} \leq \Delta(\epsilon(p_0)) \leq \Delta(\epsilon)$. This implies that on the event $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{p \geq p_0} \sum_{2^{u(16p)} \leq t < T, t \in \mathcal{T}_p} r_t(\pi(X_t)) - r_t \leq 2\epsilon.$$

Now for $p < p_0$, by Lemma 6.6, on an event $\mathcal{H}_p$ of probability one, there exists l^p such that

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t \leq T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t(\pi^{l_p}(X_t)) \leq \frac{\epsilon}{p_0}.$$

Following the arguments in the proof of Theorem 6.12, on the event $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} \cap \bigcap_{p<p_0} \mathcal{H}_p$ of probability one, for T large enough,

$$\begin{aligned}\sum_{p < p_0} \sum_{2^{u(16p)} \leq t \leq T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t &\leq \sum_{p < p_0} \sum_{2^{u(16p)} \leq t \leq T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t(\pi^{l_p}(X_t)) \\ &\quad + \sum_{p < p_0} \sum_{2^{u(16p)} \leq t \leq T, t \in \mathcal{T}_p} r_t(\pi^{l_p}(X_t)) - r_t \\ &\leq \sum_{p < p_0} \sum_{2^{u(16p)} \leq t \leq T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t(\pi^{l_p}(X_t)) \\ &\quad + 2^{\hat{l}_1} + 2^{-i(T)} T + cp_0 T^{15/16} + \epsilon T.\end{aligned}$$

As a result,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{p < p_0} \sum_{2^{u(16p)} \leq t \leq T, t \in \mathcal{T}_p} r_t(\pi^*(X_t)) - r_t \leq 2\epsilon.$$

Combining all the estimates together, we proved that on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} \cap \bigcap_{p<p_0} \mathcal{H}_p$ of probability one,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi^*(X_t)) - r_t \leq 4\epsilon.$$

This holds for all $\epsilon > 0$. The same arguments as in the original proof conclude that the learning rule is universally consistent under $\mathbb{X}$. This ends the proof of the theorem. ■

As a summary, we generalized all results from the case of the unrestricted reward to uniformly-continuous rewards with the corresponding assumptions on action spaces.

Part II

Memory Constraints in Optimization

Chapter 7

Quadratic Memory is Necessary for Optimal Query Complexity in Convex Optimization

7.1 Introduction

We consider the canonical problem of first-order convex optimization in which one aims to minimize a convex function $f : \mathbb{R}^d \to \mathbb{R}$ with access to an oracle that for any query $\boldsymbol{x}$ returns $(f(\boldsymbol{x}), \nabla f(\boldsymbol{x}))$ the value of the function and a subgradient of f at $\boldsymbol{x}$. Arguably, this is one of the most fundamental problems in optimization, mathematical programming and machine learning.

A classical question is how many oracle queries are required to find an ϵ-approximate minimizer for any 1-Lipschitz convex functions $f : \mathbb{R}^d \to \mathbb{R}$ over the unit ball. We denote by $B_d(\boldsymbol{x}, r) = \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \leq r\}$ the ball centered in $\boldsymbol{x}$ of radius r. There exist methods that given first-order oracle access only need $\mathcal{O}(d \log 1/\epsilon)$ queries and this query complexity is worst-case optimal [NY83] when $\epsilon \ll 1/\sqrt{d}$. Known methods achieving the optimal $\mathcal{O}(d \log 1/\epsilon)$ query complexity fall in the broad class of cutting plane methods, that build upon the well-known ellipsoid method [YN76b; Sho77] which uses $\mathcal{O}(d^2 \log 1/\epsilon)$ queries. These include the inscribed ellipsoid [Tar88; Nes89], volumetric center or Vaidya's method [AV95; Vai96], approximate center-of-mass via sampling techniques [Lev65; BV04] and recent improvements [LSW15; Jia+20]. Unfortunately, all these methods suffer from at least $\Omega(d^3 \log 1/\epsilon)$ time complexity and further require storing all subgradients, or at least an ellipsoid in $\mathbb{R}^d$, therefore at least $\Omega(d^2 \log 1/\epsilon)$ bits of memory. These limitations are prohibitive for large-scale optimization, hence cutting plane methods are viewed as rather impractical and less frequently used for high-dimensional applications. On the other hand, the simplest, perhaps most commonly used and practical gradient descent requires $\mathcal{O}(1/\epsilon^2)$ queries, which is not optimal for $\epsilon \ll 1/\sqrt{d}$, but only needs $\mathcal{O}(d)$ time per query and $\mathcal{O}(d \log 1/\epsilon)$ memory.

A natural question is whether one can preserve the optimal query lower bounds from cutting-plane methods with simpler methods, for instance, inspired by gradient descent techniques. Such hope is largely motivated by the fact that in many different theoretical set-

tings, cutting plane methods have achieved state-of-the-art runtimes including semidefinite programming [Ans00; LSW15], submodular optimization [McC05; GLS12; LSW15; Jia21] or equilibrium computation [PR08; JL11]. Towards this goal, [WS19] first posed this question in terms of query complexity / memory trade-off: given a certain number of bits of memory, which query complexity is achievable? While cutting planes methods require $\Omega(d^2 \log 1/\epsilon)$ memory, gradient descent only requires storing one vector and as a result, uses $\mathcal{O}(d \log 1/\epsilon)$ memory, which is information-theoretically optimal [WS19][1]. Understanding this trade-off could pave the way for the design of more efficient methods in convex optimization.

The first result in this direction was provided in [Mar+22], where they showed that it is impossible to be both optimal in query complexity and in memory. Specifically, any potentially randomized algorithm that uses at most $d^{1.25-\delta}$ memory must make at least $\tilde{\Omega}(d^{1+4/3\delta})$ queries. Thus, a super-linear amount of memory $d^{1.25}$ is required to achieve the optimal rate of convergence (that is achieved by algorithms using more than quadratic memory). However, this leaves open the fundamental question of whether one can improve over the memory of cutting-plane methods while keeping optimal query complexity.

Question 7.1 (COLT 2019 [WS19])**.** *Is it possible for a first-order algorithm that uses at most $\mathcal{O}(d^{2-\delta})$ bits of memory to achieve query complexity $\tilde{\mathcal{O}}(d \operatorname{polylog} 1/\epsilon)$ when $d = \Omega(\log^c 1/\epsilon)$ but $d = o(1/\epsilon^c)$ for all $c > 0$?*

In this chapter, building upon the techniques introduced in [Mar+22], we provide a negative answer to this question: quadratic memory is necessary to achieve optimal query complexity with deterministic algorithms. As a result, cutting plane methods including the standard center-of-mass algorithm are Pareto-optimal up to logarithmic factors within the query complexity / memory trade-off. Our main result for convex optimization is the following.

Theorem 7.1. *For $\epsilon = 1/d^4$ and any $\delta \in [0, 1]$, a deterministic first-order algorithm guaranteed to minimize 1-Lipschitz convex functions over the unit ball with ϵ accuracy uses at least $d^{2-\delta}$ bits or makes $\tilde{\Omega}(d^{1+\delta/3})$ queries.*

A key component of cutting plane methods is that they merely rely on the subgradient information at each query to restrict the search space. As a result, these can be used to solve the larger class of feasibility problems that are essential in mathematical programming and optimization. In a feasibility problem, one aims to find an ϵ-approximation of an unknown vector $\boldsymbol{x}^\star$, and has access to a separation oracle. For any query $\boldsymbol{x}$, the separation oracle either returns a separating hyperplane $\boldsymbol{g}$ from $\boldsymbol{x}$ to $B_d(\boldsymbol{x}^\star, \epsilon)$—such that $\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{z} \rangle > 0$ for any $\boldsymbol{z} \in B_d(\boldsymbol{x}^\star, \epsilon)$—or signals that $\|\boldsymbol{x} - \boldsymbol{x}^\star\| \leq \epsilon$. This class of problems is broader than convex optimization since the negative subgradient always provides a separating hyperplane from a suboptimal query to the optimal set. Hence, feasibility and convex minimization problem are closely related and it is often the case that obtaining query lower bounds for the feasibility problem simplifies the analysis while still providing key insights for the more restrictive convex optimization problem [NY83; Nes03]. Thus, a similar fundamental question is to understand the query complexity / memory trade-off for the feasibility problem. As noted above, any lower bound for convex optimization yields the same lower bound for the feasibility problem. Here, we can significantly improve over the previous trade-off.

[1] $\Omega(d \log 1/\epsilon)$ bits of memory are already required just to represent the answer to the optimization problem.

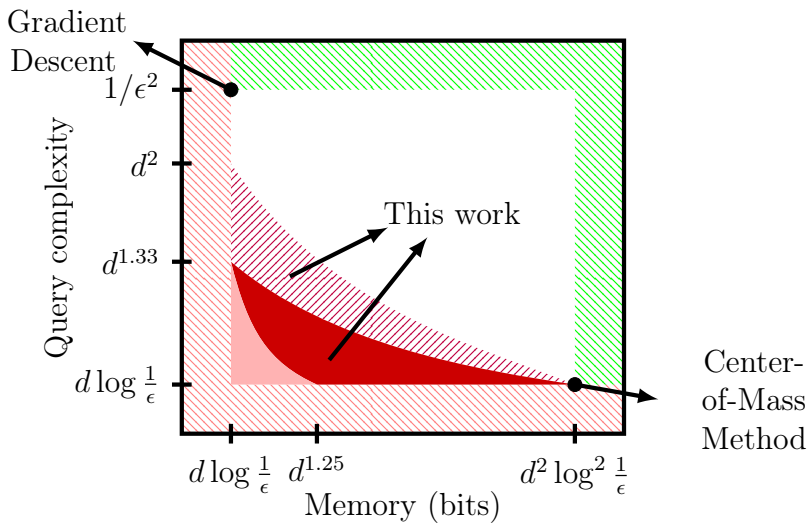

Figure 7.1: Mamroy and oracle-complexity trade-offs for minimizing 1-Lipschitz convex functions over the unit ball (adapted from [WS19; Mar+22]). The dashed pink (resp. green) region corresponds to historical information-theoretic lower bounds (resp. upper bounds) on the memory and query-complexity. The solid pink region corresponds to the recent lower bound trade-off from [Mar+22], which holds for randomized algorithms. In our work, we show that the solid red region is not achievable for any deterministic algorithm. For the feasibility problem, we also show that the dashed red region is not achievable either for any deterministic algorithm.

Theorem 7.2. *For $\epsilon = 1/(48d^2\sqrt{d})$ and any $\delta \in [0, 1]$, a deterministic algorithm guaranteed to solve the feasibility problem over the unit ball with ϵ accuracy uses at least $d^{2-\delta}$ bits of memory or makes at least $\tilde{\Omega}(d^{1+\delta})$ queries.*

7.1.1 Literature review

Recently, there has been a series of studies exploring the trade-offs between sample complexity and memory constraints for learning problems, such as linear regression [SD15; SSV19], principal component analysis (PCA) [MCJ13], learning under the statistical query model [SVW16] and other general learning problems [Bro+21; BBS22; MM17; MM18; BOY18; GRT18; KRT17; DS18; DKS19].

For parity problems that meet certain spectral (mixing) requirements, [Raz18] that an exponential number of random samples is needed if the memory is sub-quadratic. Subsequently, similar trade-offs have then been obtained for various other discrete learning problems [Raz17; MM17; KRT17; MM18; BOY18; GRT18] (finite concept class). For continuous problems, [SSV19] was the first work to show sample complexity / memory lower bounds in the case of linear regression, building upon a computation tree argument introduced in [Raz17]. They show that for accuracy $\epsilon \leq 1/d^{\mathcal{O}(\log d)}$, sub-quadratic memory algorithms require $\mathcal{O}(d \log \log 1/\epsilon)$, instead of d samples with full quadratic memory.

It should also be pointed out that [DKS19] studied linear prediction problems under the

streaming model by analyzing the *Approximate Null-Vector Problem* (ANVP). Both ANVP and the *Orthogonal Vector Game* proposed in [Mar+22] (which we build upon in this work) aim at finding vectors that lie approximately in the null space of a stream of vectors, but under different settings. A major difference is that ANVP considers a streaming setting whereas in the Orthogonal Vector Game (and the game introduced in this work), the player has access to the complete input in the beginning, then fixes a memory-constrained message based on the input.

In contrast to learning with random samples, there is limited understanding of the memory-constrained optimization and feasibility problems. [NYD83] demonstrated that, in the absence of memory constraints, finding an ϵ-approximate solution for Lipschitz convex functions requires $\Omega(d \log 1/\epsilon)$ queries, which can be achieved by the center-of-mass method using $O(d^2 \log^2 1/\epsilon)$ bits of memory. At the other extreme, gradient descent needs $\Omega(1/\epsilon^2)$ queries but only $O(d \log 1/\epsilon)$ bits of memory, the minimum memory needed to represent a solution. These two extreme cases are represented by dashed pink "impossible region" and dashed green "achievable region" in Fig. 7.1. Since then, [Mar+22] showed that there is a trade-off between memory and query for convex optimization: it is impossible to be both optimal in query complexity and memory. Their lower bound is represented by the solid pink "impossible region" in Fig. 7.1. In this chapter, we significantly improve these results to match the quadratic upper bound of cutting plane methods. Additionally, there has been recent progress in the study of query complexity for randomized algorithms [WS16; WS17], and communication complexity for convex optimization in the distributed setting [AS15; Woo+21].

On the algorithmic side, the afore-mentioned methods that achieve $O(poly(d))$ query complexity [YN76b; Sho77; Tar88; Nes89; AV95; Vai96; Lev65; BV04; LSW15; Jia+20] all require at least $\Omega(d^2 \log 1/\epsilon)$ bits of memory. There is also significant literature on memory-efficient optimization algorithms, including in particular the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [Noc80; LN89]. However, the convergence behavior for even the original BFGS on non-smooth convex objectives is still a challenging, open question [LO13].

Comparison with [Mar+22] Our proof techniques build upon those from [Mar+22]. We follow the proof strategy that they introduced to derive lower bounds for the memory/query complexity. Below, we delineate which ideas and techniques are borrowed from [Mar+22] and which are the novel elements that we introduce. Details on these proof elements are given in Section 7.2.1.

First, [Mar+22] define a class of difficult functions for convex optimization of the following form

$$\max \left\{ \|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta_0, \eta_1 \left(\max_{i \le N} \boldsymbol{v}_i^\top \boldsymbol{x} - i\gamma \right) \right\}, \tag{7.1}$$

where $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{d/2 \times d})$ is a matrix with ± 1 entries sampled uniformly, and the vectors $v_i \sim \mathcal{U}(d^{-1/2}\{\pm 1\}^d)$ are sampled independently, uniformly within the rescaled hypercube. To give intuition on this class, the term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta_0$ acts as barrier : in order to observe subgradients from the other term, one needs to use queries $\boldsymbol{x}$ that are approximately within the nullspace of $\boldsymbol{A}$. The second term $\max_{i \le N} \boldsymbol{v}_i^\top \boldsymbol{x} - i\gamma$ is the "Nemirovski" function, which

was used in previous works [Nem94; BS18; Bub+19] to obtain lower bounds in parallel convex optimization. At a high level, the limitation in the lower bounds from [Mar+22] comes from the fact that one is limited in the number N of vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ that can be used in the Nemirovski function. To resolve this issue, we introduce adaptivity within the choice of a modified Nemirovski function. At a high level, we choose the vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ depending on the queries of the algorithm which allows to fit in more terms. In turn, this allows to improve the lower bounds.

As a second step, [Mar+22] relate the optimization problem on the defined class of functions to an Orthogonal Vector Game. In this game, the goal is to find vectors that are approximately orthogonal to a matrix $\boldsymbol{A}$ with access to row queries of $\boldsymbol{A}$. The argument is as follows: because of the barrier term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta_0$, optimizing the Nemirovski function requires exploring independent directions of the nullspace of $\boldsymbol{A}$, which is performed at *informative queries*. With our new class of functions, we can adapt this logic. However, the adaptivity in the vectors $\boldsymbol{v}_i$ provides information to the learner on $\boldsymbol{A}$ in addition to the queried rows of $\boldsymbol{A}$. We therefore need to modify the game by introducing an Orthogonal Vector Game with Hints, where hints encapsulate this extra information.

For the last step, [Mar+22] give an information-theoretic argument to provide a query complexity lower bound on the defined Orthogonal Vector Game. We show that a similar argument holds for our modified game. The main added difficulty resides in bounding the information leakage from the hints, and we show that these provide no more information than the memory itself.

As a last remark, the lower bounds provided in [Mar+22] hold for randomized algorithms, while the adaptivity of our procedure only applies to deterministic algorithms.

7.1.2 Outline of the chapter

In Section 7.2, we formally define our setup and give a brief overview of our proof techniques. The complete proof of Theorem 7.1 for convex optimization is given in Section 7.3. In Section 7.4 we consider the feasibility problem and prove Theorem 7.2. Technical lemmas are proved in the appendix in Section 7.5.

7.2 Formal setup and overview of techniques

Standard results in oracle complexity give the minimal number of queries for algorithms to solve a given problem. However, this does not account for possible restrictions on the memory available to the algorithm. In this work, we are interested in the trade-off between memory and query complexity for both convex optimization and the feasibility problem. Our results apply to a large class of *memory-constrained* algorithms. We give below a general definition of the memory constraint for algorithms with access to an oracle $\mathcal{O} : \mathcal{S} \to \mathcal{R}$ taking as input a query $q \in \mathcal{S}$ and returning a response $\mathcal{O}(q) \in \mathcal{R}$.

Definition 7.1 (M-bit memory-constrained deterministic algorithm)**.** *Let $\mathcal{O} : \mathcal{S} \to \mathcal{R}$ be an oracle. An M-bit memory-constrained deterministic algorithm is specified by a query function $\psi_{query} : \{0,1\}^M \to \mathcal{S}$ and an update function $\psi_{update} : \{0,1\}^M \times \mathcal{S} \times \mathcal{R} \to \{0,1\}^M$.*

The algorithm starts with the memory state $\mathsf{Memory}_0 = 0^M$ *and iteratively makes queries to the oracle. At iteration* t*, it makes the query* $q_t = \psi_{query}(\mathsf{Memory}_{t-1})$ *to the oracle, receives the response* $r_t = \mathcal{O}(q_t)$ *then updates its memory* $\mathsf{Memory}_t = \psi_{update}(\mathsf{Memory}_{t-1}, q_t, r_t)$.

The algorithm can stop making queries at any iteration and the last query is its final output. Notice that the memory constraint applies only between each query but not for internal computations: the computation of the update ψ_{update} and the query ψ_{query} can potentially use unlimited memory. This is a rather weak memory constraint on the algorithm; a fortiori, our negative results also apply to stronger notions of memory-constrained algorithms. In Definition 7.1, we ask the query and update functions to be time-invariant, which in our context is without loss of generality: any M-bit algorithm using T queries with time-dependent query and update functions [WS19; Mar+22] can be turned into an $(M + \lceil \log T \rceil)$-bit time-invariant algorithm by storing the iteration number t as part of the memory. The query lower bounds we provide are at most $T \leq poly(d)$. Hence, an additional $\log T = O(\log d)$ bits to the memory size M does not affect our main results, Theorems 7.1 and 7.2.

We use the above described framework to study the interplay between query complexity and memory for two fundamental problems in optimization and machine learning.

Convex optimization. We first consider convex optimization in which one aims to minimize a 1-Lipschitz convex function $f : B_d(\mathbf{0}, 1) \to \mathbb{R}$ on the unit ball. The goal is to output a point $\tilde{\boldsymbol{x}} \in B_d(\mathbf{0}, 1)$ such that $f(\tilde{\boldsymbol{x}}) \leq \min_{\boldsymbol{x} \in B_d(\mathbf{0},1)} f(\boldsymbol{x}) + \epsilon$, referred to as ϵ-approximate solutions. The optimization algorithm has access to a first order oracle $\mathcal{O}_{CO} : B_d(\mathbf{0}, 1) \to \mathbb{R} \times \mathbb{R}^d$, which for any query $\boldsymbol{x}$ returns the couple $(f(\boldsymbol{x}), \partial f(\boldsymbol{x}))$ where $\partial f(\boldsymbol{x})$ is a subgradient of f at the query point $\boldsymbol{x}$.

Feasibility problem. Second, we consider the trade-off between memory and query complexity for the feasibility problem, where the goal is to find an element $\tilde{\boldsymbol{x}} \in Q$ for a convex set $Q \subset B_d(\mathbf{0}, 1)$. Instead of a first-order oracle, the algorithm has access to a separation oracle $\mathcal{O}_F : B_d(\mathbf{0}, 1) \to \{\mathsf{Success}\} \cup \mathbb{R}^d$. For any query $\boldsymbol{x} \in B_d(\mathbf{0}, 1)$, the separation oracle either returns $\mathsf{Success}$ reporting that $\boldsymbol{x} \in Q$, or provides a separating vector $\boldsymbol{g} \in \mathbb{R}^d$, i.e., such that for all $\boldsymbol{x}' \in Q$,

$$\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}' \rangle > 0.$$

We say that an algorithm solves the feasibility problem with accuracy $\epsilon > 0$ if it can solve any feasibility problem for which the successful set contains a ball of radius ϵ, i.e., such that there exists $\boldsymbol{x}^\star \in B_d(\mathbf{0}, 1)$ satisfying $B_d(\boldsymbol{x}^\star, \epsilon) \subset Q$.

The feasibility problem is at least as hard as convex optimization in the following sense: an algorithm that solves the feasibility problem with accuracy ϵ/L can be used to solve L-Lipschitz convex optimization problems by feeding the subgradients from first-order queries to the algorithm as separating hyperplanes. Alternatively, from any 1-Lipschitz function f one can derive a feasibility problem, where the feasibility set is $Q = \{\boldsymbol{x} \in B_d(\mathbf{0}, 1), f(\boldsymbol{x}) \leq f^\star + \epsilon\}$ and the separating oracle at $\boldsymbol{x} \notin Q$ is a subgradient $\partial f(\boldsymbol{x})$ at $\boldsymbol{x}$.

Remark 7.1. *Although we consider the case of constrained optimization, one can efficiently reduce the problem of approximate Lipschitz convex optimization over the unit ball to unconstrained approximate Lipschitz convex optimization [Mar+22]. Hence, our results also apply to the latter setting at the expense of losing poly(d) factors in the necessary accuracy ϵ in Theorem 7.1. For the feasibility problem, there is no loss, Theorem 7.2 applies directly for the unconstrained feasibility setting.*

7.2.1 Overview of proof techniques and innovations

We prove the two main Theorems 7.1 and 7.2 with similar techniques, hence for conciseness, we only give here the main ideas used to derive lower bounds for convex optimization. Although our proof borrows techniques from [Mar+22], we introduce key innovations involving adaptivity to improve the lower bounds up to the maximum quadratic memory for deterministic algorithms—up to logarithmic factors. We recall, however, that the bounds in [Mar+22] hold for randomized algorithms as well. In the proofs, we aim to optimize the dependence of the parameters in d. Constants, however, are not necessarily optimized.

An adaptive optimization procedure. At the high level, we design an *optimization procedure* which for any algorithm constructs a hard family of convex functions adaptively on its queries. To be precise, the procedure constructs functions from the following family of convex functions with appropriately chosen parameters $\eta, \gamma_1, \gamma_2, p_{max}, l_p, \delta$:

$$F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) = \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \eta \boldsymbol{v}_0^\top \boldsymbol{x}, \eta\left(\max_{p\leq p_{max}, l\leq l_p} \boldsymbol{v}_{p,l}^\top \boldsymbol{x} - p\gamma_1 - l\gamma_2\right)\right\}. \tag{7.2}$$

We take $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n\times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$ uniformly sampled in the beginning, where $\mathcal{D}_\delta \subset \mathcal{S}^{d-1}$ is a (finite) discretization of the sphere. The first term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta$ acts as a barrier term: in order to observe subgradients from the other terms, one needs the query $\boldsymbol{x}$ to satisfy $\|\boldsymbol{A}\boldsymbol{x}\|_\infty \leq 2\eta$. These are called *informative queries* as introduced in [Mar+22]. Hence, informative queries must lie approximately in the orthogonal space to the lines of $\boldsymbol{A}$. The second term $\eta \boldsymbol{v}_0^\top \boldsymbol{x}$ ensures that queries with low objective (in particular with objective at most $-\eta\gamma_1/2$) have norm bounded away from 0. Thus, these queries, once renormalized, will still belong approximately to the nullspace of $\boldsymbol{A}$ denoted $Ker(\boldsymbol{A})$.

The adaptivity to the algorithm is captured in the third term, which is constructed along the optimization process. This construction proceeds by periods $p = 1, 2, \ldots, p_{max}$ designed so that during each period p, the algorithm is forced to visit a subspace of $Ker(\boldsymbol{A})$ of dimension k. To do so, we iteratively construct vectors $\boldsymbol{v}_{p,1}, \ldots \boldsymbol{v}_{p,l_p}$ as follows. Suppose that at the beginning of step t of period p, one has defined vectors $\boldsymbol{v}_{p,1}, \ldots, \boldsymbol{v}_{p,l}$.

- The procedure first evaluates the explored subspace of the algorithm during this period. In practice, the procedure keeps in memory *exploratory* queries $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}}$ during period p up to time t. The exploratory subspace is then $Span(\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}})$.
- If a query with a sufficiently low objective is queried, we sample a new vector $\boldsymbol{v}_{p,l+1}$ which is approximately orthogonal to the exploratory subspace. The corresponding new term in the objective is $\boldsymbol{v}_{p,l+1}^\top \boldsymbol{x} - p\gamma_1 - (l+1)\gamma_2$.

Once this new term is added to the objective, the algorithm is constrained to make queries with an additional component along the direction $-\boldsymbol{v}_{p,l+1}$. Since this vector is approximately orthogonal to all previous queries, this forces the algorithm to query vectors linearly independent from all previous queries in period p. The period then ends once the dimension of the exploratory subspace reaches k, having defined l_p vectors $\boldsymbol{v}_{p,1}, \dots, \boldsymbol{v}_{p,l_p}$. As discussed above, the exploratory subspace must increase dimension for any additional such vector. Thus, after $l_p \leq k$ vectors, period p ends.

The constructed family of convex functions in Eq (7.2) is similar to the family described in Eq (7.1) that were considered in [Mar+22]. However, by sampling the vectors $\boldsymbol{v}_{p,l}$ adaptively, the *optimization procedure* is able to fit in more terms, thereby providing a significant improvement in the lower bounds.

Benefits of adaptivity. We now expand on how the adaptive terms allow improving the lower bound of [Mar+22] to match the quadratic upper bound of cutting plane methods. The limitation in the functions of the form Eq (7.1) comes from the fact that the offset in the Nemirovski function is $\gamma = \Omega(\sqrt{k \log d/d})$. This offset is necessary to ensure that with high probability, 1: subgradients $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ are discovered exactly in this order and 2: that any query which visits a new vector $\boldsymbol{v}_i$ must not lie in the subspace formed by the last k last informative vectors. Indeed, for the last claim, from high-dimensional concentration, for a random unit vector $\boldsymbol{v}$ and a k dimensional subspace E, $\|P_E(\boldsymbol{v})\| = \Theta(\sqrt{k \log d/d})$. This offset is not necessary for our procedure, since by construction, at each period, a k-dimensional subspace of $Ker(\boldsymbol{A})$ is forced to be explored. As a result, we can take $\gamma_1 = \Theta(\sqrt{\log d/d})$. This offset is still necessary to ensure that vectors $\boldsymbol{v}_{p,l}$ are discovered in their order of construction (lexicographic order on (p, l)) with high probability.

An Orthogonal Vector Game with Hints. The next step of the proof involves linking the optimization of the above-mentioned constructed functions with an Orthogonal Vector Game with Hints. Similarly to the game introduced by [Mar+22], the goal for the player is to find k linearly-independent vectors approximatively in $Ker(\boldsymbol{A})$. To do so, the player can access an M-bit message $\mathsf{Message}$ and make m queries, where $M = ckd$ for a small constant $c > 0$. In the game introduced by [Mar+22], the queries are lines of the matrix $\boldsymbol{A}$. They then show that to find k dimensions of $Ker(\boldsymbol{A})$, where $\boldsymbol{A}$ is taken uniformly at random $\boldsymbol{A} \sim \{\pm 1\}^{d/2 \times d}$, (nearly) all the lines of $\boldsymbol{A}$ must be queried. The argument is information-theoretic: each new dimension of $Ker(\boldsymbol{A})$ must be (approximately) orthogonal to all lines of $\boldsymbol{A}$. Hence, this provides additional mutual information $O(k)$ for every line of $\boldsymbol{A}$, including the $d/2 - m$ lines that were not observed through queries. This extra information on $\boldsymbol{A}$ can only be explained by the message, which has M bits. Hence, $M \geq O(k)(d/2 - m)$. Setting the constant $c > 0$ appropriately, this shows that $m = \Omega(d)$.

In our case, the optimization procedure ensures that the algorithm needs to explore k dimensions of $Ker(\boldsymbol{A})$ in each period. However, each query yields a response from the optimization oracle that can either be a line of $\boldsymbol{A}$ (corresponding to the term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta$ of Eq (7.2)) or $\boldsymbol{v}_0$ (term $\eta \boldsymbol{v}_0^\top \boldsymbol{x}$ of Eq (7.2)), or previously defined vectors $\boldsymbol{v}_{p,l,p}$. Since the vectors $\boldsymbol{v}_{p',l'}$ have been constructed adaptively on the queries of the algorithm, which themselves may depend on lines of $\boldsymbol{A}$, during a period p, responses $\boldsymbol{v}_{p',l'}$ for $p' < p$ are a source

of information leakage for $\boldsymbol{A}$ from previous periods. As a result, the query lower bound on the game introduced by [Mar+22] is not sufficient for our purposes. Instead, we introduce an Orthogonal Vector Game with Hints, where hints correspond exactly to these vectors $\boldsymbol{v}_{p',l'}$ from previous periods. Informally, the game corresponds to a simulation of one of the periods of the optimization procedure: for each query $\boldsymbol{x}$, the oracle returns the subgradient that would have been returned in the optimization procedure, up to minor details.

Bounding the information leakage. Once the link is settled, the goal is to prove lower bounds on the number of queries needed to solve the Orthogonal Vector Game with Hints. The main difficulty is to bound the information leakage from these hints. We recall that hints are of the form $\boldsymbol{v}_{p',l'}$, which have been constructed adaptively on the queries of the algorithm during period p'. In particular, these contain information on the lines of $\boldsymbol{A}$ queried during period $p' < p$, which may be complementary with those queried during period p. If this total information leakage through the hints yields a mutual information with $Ker(\boldsymbol{A})$ significantly higher than that of the M bits of Message, obtained lower bounds cannot possibly reflect any trade-off with memory constraints. It is therefore essential to obtain information leakage at most $\tilde{\mathcal{O}}(M) = \tilde{\mathcal{O}}(dk)$.

To solve this issue, we introduce a discretization $\mathcal{D}_\delta$ of the unit sphere where the vectors $\boldsymbol{v}_{p,l}$ take value. Next, we show that each individual vector $\boldsymbol{v}_{p',l'}$ from previous periods can only provide information $\tilde{\mathcal{O}}(k)$ on the matrix $\boldsymbol{A}$. To have an intuition on this, note that for any (at most) k vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$, the volume of the subset of the unit sphere S^{d-1} of vectors approximately orthogonal to $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$, say $S(\boldsymbol{x}_1, \dots, \boldsymbol{x}_k) = \{\boldsymbol{y} \in S^{d-1} : |\boldsymbol{y}^\top \boldsymbol{x}_i| \leq d^{-3}, i \leq k\}$ is $q_k = \Omega(1/d^{3k})$. Hence, since the vector $\boldsymbol{v}$ is roughly taken uniformly at random within $\mathcal{D}_\delta \cap S(\boldsymbol{x}_1, \dots, \boldsymbol{x}_k)$, we can show that the mutual information of $\boldsymbol{v}$ with the initial vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ is at most $\mathcal{O}(-\log q_k) = \mathcal{O}(k \log d)$. As a result, even if $m = d$, the total information leakage through the vectors $\boldsymbol{v}_{p',l'}$ from previous periods, is at most $\mathcal{O}(kd \log d)$. The formal proof involves an anti-concentration bounds on the distance of a random unit vector to a linear subspace of dimension k, as well as a more involved discretization procedure than the one presented above. In summary, by introducing adaptive functions through the optimization procedure, we show that the same memory-sample trade-off holds for the Orthogonal Vector Game with Hints and the game without hints introduced in [Mar+22], up to logarithmic factors.

7.3 Memory-constrained convex optimization

To prove our results we need to use discretizations of the unit sphere S^{d-1}, which we construct by first partitioning S^{d-1} into $N(\delta) = (\mathcal{O}(1)/\delta)^d$ regions of equal area and diameter at most δ, i.e. $\mathcal{V}_\delta = \{V_i(\delta), i \in [N(\delta)]\}$. Here $\delta > 0$ is taken as parameter. The existence of this construcition is guaranteed by the following lemma.

Lemma 7.1 ([FS02] Lemma 21)**.** *For any $0 < \delta < \pi/2$, the sphere S^{d-1} can be partitioned into $N(\delta) = (\mathcal{O}(1)/\delta)^d$ equal volume cells, each of diameter at most δ.*

Then we take one point as the representative of each region, i.e. $\mathcal{D}_\delta = \{\boldsymbol{b}_i(\delta), i \in [N(\delta)]\} \subset S^{d-1}$, where for all $i \in [N(\delta)]$, $\boldsymbol{b}_i(\delta) \in V_i(\delta)$. With these notations we define the

discretization function ϕ_δ such that for any $\boldsymbol{x} \in S^{d-1}$, $\phi_\delta(\boldsymbol{x}) = \boldsymbol{b}_i(\delta)$ where $\boldsymbol{x} \in V_i(\delta)$.

7.3.1 Definition of the difficult class of optimization problems

In this section we present the class of functions that we use to prove our lower bounds. Throughout the chapter, we pose $n = \lceil d/4 \rceil$. We first define some useful functions. For any $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, we define $\boldsymbol{g}_{\boldsymbol{A}}$ as follows

$$\boldsymbol{g}_{\boldsymbol{A}}(\boldsymbol{x}) = \boldsymbol{a}_{i_{\min}}, \qquad i_{\min} = \min\{i \in [n], |\boldsymbol{a}_i^\top \boldsymbol{x}| = \|\boldsymbol{A}\boldsymbol{x}\|_\infty\}.$$

With this function we can define a subgradient function for $\boldsymbol{x} \mapsto \|\boldsymbol{A}\boldsymbol{x}\|_\infty$,

$$\tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}) = \epsilon \boldsymbol{g}_{\boldsymbol{A}}(\boldsymbol{x}), \qquad \epsilon = sign(\boldsymbol{g}_{\boldsymbol{A}}(\boldsymbol{x})^\top \boldsymbol{x}).$$

We are now ready to introduce the class of functions which we use for our lower bounds. These are of the following form.

$$F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) = \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \eta \boldsymbol{v}_0^\top \boldsymbol{x}, \eta\left(\max_{p \le p_{max}} \max_{l \le l_p} \boldsymbol{v}_{p,l}^\top \boldsymbol{x} - p\gamma_1 - l\gamma_2\right)\right\}.$$

Here, $\boldsymbol{A} \in \{\pm 1\}^{n \times d}$ is a matrix. Also, $\boldsymbol{v}_0$ and the terms $\boldsymbol{v}_{p,l}$ are vectors in $\mathbb{R}^d$. More precisely, these vectors will lie in the discretization $\mathcal{D}_\delta$ for $\delta = 1/d^3$. We postpone the definition of p_{max} and l_p for $p \le p_{max}$. Last, we use the following choice for the remaining parameters: $\eta = 2/d^3$, $\gamma_1 = 12\sqrt{\frac{\log d}{d}}$ and $\gamma_2 = \frac{\gamma_1}{4d}$. For convenience, we also define the functions

$$\begin{aligned} F_{\boldsymbol{A}}(\boldsymbol{x}) &= \max\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \eta \boldsymbol{v}_0^\top \boldsymbol{x}\} \\ F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}) &= \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \eta \boldsymbol{v}_0^\top \boldsymbol{x}, \eta\left(\max_{(p',l') \le_{lex} (p,l), l' \le l_{p'}} \boldsymbol{v}_{p',l'}^\top \boldsymbol{x} - p'\gamma_1 - l'\gamma_2\right)\right\}, \end{aligned}$$

with the convention $F_{\boldsymbol{A},\boldsymbol{v},1,0} = F_{\boldsymbol{A}}$. The functions $F_{\boldsymbol{A},\boldsymbol{v},p,l}$ will encapsulate the current state of the function to be minimized: it will be updated adaptively on the queries of the algorithm. We also define a subgradient function for $F_{\boldsymbol{A},\boldsymbol{v},p,l}$ by first favoring lines of $\boldsymbol{A}$, then vectors from $\boldsymbol{v}$ in case of ties, as follows,

$$\partial F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}) = \begin{cases} \tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t) & \text{if } F_{\boldsymbol{A},\boldsymbol{v},l,p}(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \\ \eta \boldsymbol{v}_0 & \text{otherwise and if } F_{\boldsymbol{A},\boldsymbol{v},l,p}(\boldsymbol{x}) = \eta \boldsymbol{v}_0^\top \boldsymbol{x}, \\ \eta \boldsymbol{v}_{p,l} & \text{otherwise and if } (p,l) = \arg\max_{(p',l') \le_{lex}(p,l)} \boldsymbol{v}_{p',l'}^\top \boldsymbol{x} - p'\gamma_1 - l'\gamma_2. \end{cases}$$

In the last case, ties are broken by lexicographic order. We also pose $\partial F_{\boldsymbol{A},\boldsymbol{v}} = \partial F_{\boldsymbol{A},\boldsymbol{v},p_{max},l_{p_{max}}}$.

We consider a so-called *optimization procedure* described in Procedure 7.1, which will construct the sequence of vectors $\boldsymbol{v} = (\boldsymbol{v}_{p,l})$ adaptively on the responses of the considered algorithm. Throughout this section, we use a parameter $1 \le k \le d/3 - 1$ — which will be taken as $k = \tilde{\Theta}(M/d)$ where M is the memory of the algorithm — and let p_{max} be the largest number which satisfies the following constraint.

$$p_{max} \le \min\{c_{d,1}(d-1)/k,\ c_{d,2}(d/k)^{1/3} - 1\}, \tag{7.3}$$

Input: d, k, p_{max}, algorithm alg

Part 1: Procedure to adaptively construct $\boldsymbol{v}$;

Sample $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n\times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$.;
Initialize the memory of alg to $\mathbf{0}$ and let $p = 1$, $r = l = 0$.;
for $t \geq 1$ **do**
  if $t > d^2$ **then** Set $(P, L) = (p, l)$ and break the **for** loop ;
  Run alg with current memory to obtain a query $\boldsymbol{x}_t$;
  if $F_{\boldsymbol{A}}(\boldsymbol{x}_t) > \eta$ **then** `// Non-informative query`
    return $(\|\boldsymbol{A}\boldsymbol{x}_t\|_\infty - \eta, \tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t))$ as response to alg.
  else `// Informative query`
    if $r \leq k-1$ *and* $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) \leq -\eta\gamma_1/2$ *and* $\|P_{Span(\boldsymbol{x}_{i_{p,r'}}, r'\leq r)^\perp}(\boldsymbol{x}_t)\|/\|\boldsymbol{x}_t\| \geq \frac{\gamma_2}{4}$ **then**
      Set $i_{p,r+1} = t$ and increment $r \leftarrow r+1$.
    if $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) < -\eta(p\gamma_1 + l\gamma_2 + \gamma_2/2)$ *and* $r < k$ **then**
      Compute Gram-Schmidt decomposition $\boldsymbol{b}_{p,1}, \ldots, \boldsymbol{b}_{p,r}$ of $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}}$.;
      Sample $\boldsymbol{y}_{p,l+1}$ uniformly on $\mathcal{S}^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p,r'}^\top \boldsymbol{z}| \leq d^{-3}, \forall r' \leq r\}$.;
      Define $\boldsymbol{v}_{p,l+1} = \phi_\delta(\boldsymbol{y}_{p,l+1})$ and increment $l \leftarrow l+1$.
    else if $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) < -\eta(p\gamma_1 + l\gamma_2 + \gamma_2/2)$ *and* $p+1 \leq p_{max}$ **then**
      Set $l_p = l$ and $i_{p+1,1} = t$.;
      Compute the Gram-Schmidt decomposition $\boldsymbol{b}_{p+1,1}$ of $\boldsymbol{x}_{i_{p+1,1}}$.;
      Sample $\boldsymbol{y}_{p+1,1}$ uniformly on $\mathcal{S}^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p+1,1}^\top \boldsymbol{z}| \leq d^{-3}\}$.;
      Define $\boldsymbol{v}_{p+1,1} = \phi_\delta(\boldsymbol{y}_{p+1,1})$, increment $p \leftarrow p+1$ and reset $l = r = 1$.
    else if $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) < -\eta(p\gamma_1 + l\gamma_2 + \gamma_2/2)$ **then**`// End of the construction`
      Set $l_{p_{max}} = l$, $i_{p_{max}+1,1} = t$.;
      Set $(P, L) = (p_{max}, l)$ and break the **for** loop.
    return $(F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t), \boldsymbol{\partial} F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t))$ as response to alg.
end

Part 2: Procedure once $\boldsymbol{v}$, P, L are constructed;

for $t' \geq t$ **do return** $(F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_{t'}), \partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_{t'}))$ as response to the query $\boldsymbol{x}_{t'}$;

Procedure 7.1: The optimization procedure for algorithm alg

where $c_{d,1} = 1/(90^2 \log^2 d)$ and $c_{d,2} = 1/(81 \log^{2/3} d)$.

The optimization procedure is described in Procedure 7.1. First, we sample independently $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n\times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$. The matrix $\boldsymbol{A}$ and vector $\boldsymbol{v}_0$ are then fixed for the rest of the learning procedure. Next, we describe the adaptive procedure to return subgradients. It proceeds by periods, until p_{max} periods are completed, unless the total number of iterations reaches d^2, in which case the construction procedure ends as well. First, we say that a query is informative if $F_{\boldsymbol{A}}(\boldsymbol{x}) \leq \eta$. The procedure proceeds by periods $p \in [p_{max}]$ and in each period constructs the vectors $\boldsymbol{v}_{p,1}, \ldots, \boldsymbol{v}_{p,k}$ iteratively. We are now ready to describe the procedure at time t when the new query $\boldsymbol{x}_t$ is queried. Let $p \geq 1$ be the index of the current period and $\boldsymbol{v}_{p,1}, \ldots, \boldsymbol{v}_{p,l}$ be the vectors of this period constructed so far: the first period is $p = 1$ and we allow $l = 0$ here. As will be seen in the construction, we always have

$l \geq 1$ except at the very beginning for which we use the notation $F_{\boldsymbol{A},\boldsymbol{v},1,0} = F_{\boldsymbol{A}}$. Together with these vectors, the oracle keeps in memory indices $i_{p,1}, \ldots, i_{p,r}$ with $r \leq k$ of *exploratory* queries. The constructed vectors from previous periods are $\boldsymbol{v}_{p',l'}$ for $p' < p$ and $l' \leq l_{p'}$.

1. If $\boldsymbol{x}_t$ is not informative, i.e. $F_{\boldsymbol{A}}(\boldsymbol{x}_t) > \eta$, then procedure returns $(\|\boldsymbol{A}\boldsymbol{x}_t\|_\infty - \eta, \tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t))$.

2. Otherwise, we follow the next steps. If $r \leq k-1$ and

$$F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) \leq -\frac{\eta\gamma_1}{2} \qquad \text{and} \qquad \frac{\|P_{Span(\boldsymbol{x}_{i_{p,r'}}, r' \leq r)^\perp}(\boldsymbol{x}_t)\|}{\|\boldsymbol{x}_t\|} \geq \frac{\gamma_2}{4},$$

   we set $i_{p,r+1} = t$ and increment r. In this case, we say that $\boldsymbol{x}_t$ is *exploratory*. Next,

   (a) Recalling that $F_{\boldsymbol{A},\boldsymbol{v},p,l}$ is constructed so far, if $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) \geq \eta(-p\gamma_1 - l\gamma_2 - \gamma_2/2)$, we do not do anything.

   (b) Otherwise, and if $r < k$, let $\boldsymbol{b}_{p,1}, \ldots, \boldsymbol{b}_{p,r}$ be the result from the Gram-Schmidt decomposition of $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}}$. Then, let $\boldsymbol{y}_{p,l+1}$ be a sample of the distribution obtained via $\boldsymbol{y}_{p,l+1} \sim \mathcal{U}(S^{d-1} \cap \left\{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p,r'}^\top \boldsymbol{z}| \leq \frac{1}{d^3}, \forall r' \leq r\right\})$. We then pose $\boldsymbol{v}_{p,l+1} = \phi_\delta(\boldsymbol{y}_{p,l+1})$. Having defined this new vector, we increment l.

   (c) Otherwise, if $r = k$, this ends period p. We write the total number of vectors defined during period p as $l_p := l$. If $p+1 \leq p_{max}$, period $p+1$ starts from $t = i_{p+1,1}$. Similarly to above, let $\boldsymbol{b}_{p+1,1}$ be the result of the Gram-Schmidt procedure on $\boldsymbol{x}_{p+1,1}$, and we sample $\boldsymbol{y}_{p+1,1}$ according to a uniform distribution $\boldsymbol{y}_{p+1,1} \sim \mathcal{U}(S^{d-1} \cap \left\{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p+1,1}^\top \boldsymbol{z}| \leq \frac{1}{d^3}\right\})$. Then, we pose $\boldsymbol{v}_{p+1,1} = \phi_\delta(\boldsymbol{y}_{p+1,1})$, increment p, and reset $l = r = 1$.

   After these steps, with the current values of p and l, we return to the algorithm $(F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t), \partial F_{\boldsymbol{A},\boldsymbol{v},l,p}(\boldsymbol{x}_t))$.

If we finish the last period $p = p_{max}$, or if we reach a total number of iterations d^2, the construction phase of the function ends. In both cases, let us denote by P, L the last defined period and vector $\boldsymbol{v}_{P,L}$. In particular, we have $p \leq p_{max}$ From now on, the final function to optimize is $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ and the oracle is a standard first-order oracle for this function, using the subgradient function $\partial F_{\boldsymbol{A},\boldsymbol{v},P,L}$.

7.3.2 Sketch of proof for Theorem 7.1

Throughout the proof of the main results, we will use concentration bounds relegated to the appendix in Section 7.5.1. We first relate Procedure 7.1 to the standard convex optimization problem and prove query lower bounds under memory constraints for this procedure. Before doing so, we formally define what we mean by solving this optimization procedure.

Definition 7.2. *Let alg be an algorithm for convex optimization. We say that an algorithm alg is successful for the optimization procedure with probability $q \in [0,1]$ and accuracy $\epsilon > 0$, if taking $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n \times d})$, running alg with the responses given by the procedure, and denoting by $\boldsymbol{x}^\star(alg)$ the final answer returned by alg, with probability at least q over the randomness of $\boldsymbol{A}$ and of the procedure, one has*

$$F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}^\star(alg)) \leq \min_{\boldsymbol{x} \in B_d(\boldsymbol{0},1)} F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}) + \epsilon.$$

The optimization procedure is designed such that with probability at least $1-C\sqrt{\log d}/d^2$, the procedure returns responses that are consistent with a first-order oracle of the function $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ where $\boldsymbol{v}_{P,L}$ is the last vector to have been defined.

Proposition 7.1. *Let $\boldsymbol{A} \in \{\pm 1\}^{n\times d}$ and $\boldsymbol{v}_0 \in \mathcal{D}_\delta$. On an event $\mathcal{E}$ of probability at least $1 - C\sqrt{\log d}/d^2$ on the randomness of the procedure for some universal constant $C > 0$, all responses of the optimization procedure are consistent with a first-order oracle for the function $F_{\boldsymbol{A},\boldsymbol{v},P,L}$: for any $t \geq 1$, if $(f_t, \boldsymbol{g}_t)$ is the response of the procedure at time t for query $\boldsymbol{x}_t$, then $f_t = F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t)$ and $\boldsymbol{g}_t = \partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t)$.*

Now observe that for any constructed vectors $\boldsymbol{v}$, the function $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ is $\sqrt{d}$-Lipschitz. As a result, if there exists an algorithm for convex optimization that guarantees precision ϵ for 1-Lipschitz functions, by rescaling, there exists an algorithm alg which is successful for the optimization procedure with probability $1 - C\sqrt{\log d}/d^2$ and precision $\epsilon\sqrt{d}$. In the next proposition, we show that to be successful, such an algorithm needs to properly define the complete function $F_{\boldsymbol{A},\boldsymbol{v}}$, i.e., to complete all periods until p_{max}.

Proposition 7.2. *Let alg be a successful algorithm for the optimization procedure with probability $q \in [0, 1]$ and precision $\eta/(2\sqrt{d})$. Suppose that alg performs at most d^2 queries during the optimization procedure. Then when running alg with the responses of the optimization procedure, alg succeeds and ends the period p_{max} with probability at least $q - C\sqrt{\log d}/d$ for some universal constant $C > 0$.*

Next, we introduce an Orthogonal Vector Game with Hints, Game 7.2, where the main difference with the game introduced in [Mar+22] is that the player can provide additional hints. Using Proposition 7.2, we prove that solving the optimization procedure implies solving Game 7.2.

Proposition 7.3. *Let $m \leq d$. Suppose that there is an M-bit algorithm that is successful for the optimization procedure with probability q for accuracy $\epsilon = \eta/(2\sqrt{d})$ and uses at most mp_{max} queries. Then, there is an algorithm for Game 7.2 for parameters $(d, k, m, M, \alpha = \frac{2\eta}{\gamma_1}, \beta = \frac{\gamma_2}{4})$, for which the Player wins with probability at least $q - C\sqrt{\log d}/d$ for some universal constant $C > 0$.*

Last, we give a $m = \tilde{\Omega}(d)$ query lower bound for Game 7.2.

Proposition 7.4. *Let $k \geq 20\frac{M+3d\log(2d)+1}{c_H n}$. And let $0 < \alpha, \beta \leq 1$ such that $\alpha(\sqrt{d}/\beta)^{5/4} \leq \frac{1}{2}$. If the Player wins the Orthogonal Vector Game with Hints (Game 7.2) with probability at least $1/2$, then $m \geq \frac{c_H}{8(30\log d + c_H)}d$.*

Putting everything together, we prove our main result.

Proof of Theorem 7.1 We set $n = \lceil d/4 \rceil$ and $k = \lceil 20\frac{M+3d\log(2d)+1}{c_H n} \rceil$. By Proposition 7.1, with probability at least $1 - C\sqrt{\log d}/d^2$, the procedure is consistent with a first-order oracle for convex optimization. Hence, since the functions $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ are $\sqrt{d}$-Lipschitz, any M-bit algorithm guaranteed to solve convex optimization within accuracy $\epsilon = \eta/(2d) = 1/d^4$ for 1-Lipschitz functions, yields an algorithm that is successful for the optimization procedure with probability at least $1 - C\sqrt{\log d}/d^2$ and precision $\epsilon\sqrt{d} = \eta/(2\sqrt{d})$. Suppose that it uses

Input: d, k, m, M, α, β

Oracle: Set $n \leftarrow \lfloor d/4 \rfloor$, sample $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n \times d})$.;

Player: Observe $\boldsymbol{A}$;

for $l \in [d]$ **do**

  Player: Based on $\boldsymbol{A}$ and any previous queries and responses, submit at most k vectors $\boldsymbol{x}_{l,1}, \ldots, \boldsymbol{x}_{l,r_l}$.;

  Oracle: Perform the Gram-Schmidt decomposition $\boldsymbol{b}_{l,1}, \ldots, \boldsymbol{b}_{l,r_l}$ of $\boldsymbol{x}_{l,1}, \ldots, \boldsymbol{x}_{l,r_l}$. Then, sample a vector $\boldsymbol{y}_l \in S^{d-1}$ according to a uniform distribution $\mathcal{U}(S^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : \forall r \leq r_l, |\boldsymbol{b}_{l,r}^\top \boldsymbol{z}| \leq d^{-3}\})$. As response to the query, return $\boldsymbol{v}_l = \phi_\delta(\boldsymbol{y}_l)$ to the player.

end

Player: Based on $\boldsymbol{A}$, all previous queries and responses, store an M-bit message Message.;

Player: Based on $\boldsymbol{A}$, all previous queries and responses, submit a function $\boldsymbol{g} : B_d(\boldsymbol{0}, 1) \to (\{\boldsymbol{a}_j, j \leq n\} \cup \{\boldsymbol{v}_l, l \leq d\}) \times [d^2]$ to the Oracle.

for $i \in [m]$ **do**

  Player: Based on Message, any previous queries $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}$ and responses $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{i-1}$ from this loop phase, submit a query $\boldsymbol{x}_i \in \mathbb{R}^d$.;

  Oracle: As the response to query $\boldsymbol{z}_i$, return $\boldsymbol{g}_i = \boldsymbol{g}(\boldsymbol{z}_i)$.

end

Player: Based on all queries and responses from this phase $\{\boldsymbol{z}_i, \boldsymbol{g}_i, i \in [m]\}$, and on Message, return some vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ to the oracle.;

The player wins if the returned vectors have unit norm and satisfy for all $i \in [k]$

1. $\|\boldsymbol{A}\boldsymbol{y}_i\|_\infty \leq \alpha$
2. $\|P_{Span(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1})^\perp}(\boldsymbol{y}_i)\|_2 \geq \beta$.

Game 7.2: Orthogonal Vector Game with Hints

at most Q queries. Then, by Proposition 7.3, there is a strategy for Game 7.2 for parameters $(d, k, \lceil Q/p_{max} \rceil + 1, M, \alpha = \frac{2\eta}{\gamma_1}, \beta = \frac{\gamma_2}{4})$ in which the Player wins with probability at least $1 - C'\sqrt{\log d}/d$. For d large enough, this probability is at least $1/2$. Further, $\frac{2\eta}{\gamma_1}\left(\frac{4\sqrt{d}}{\gamma_2}\right)^{5/4} \leq \frac{(4/3)^{5/4}}{3}\eta d^3 \leq \frac{1}{2}$. Hence, by Proposition 7.4, one has $\lceil Q/p_{max} \rceil + 1 \geq \frac{c_H}{8(30 \log d + c_H)} d$. Because one has $p_{max} = \Theta((d/k)^{1/3} \log^{-2/3} d)$, this implies

$$Q = \Omega\left(\frac{(d/k)^{1/3} d}{\log^{5/3} d}\right) = \Omega\left(\frac{d^{5/3}}{(M + \log d)^{1/3} \log^{5/3} d}\right).$$

In particular, if $M = d^{1+\delta}$ for $\delta \in [0, 1]$, the number of queries is $Q = \tilde{\Omega}(d^{1+(1-\delta)/3})$. ■

In the rest of this section, we give all the remaining details and proofs leading up to Theorem 7.1.

7.3.3 Properties and validity of the optimization procedure

We begin with a simple lemma showing that during each period p at most $l_p \leq k$ vectors $\boldsymbol{v}_{p,1}, \ldots, \boldsymbol{v}_{p,l_p}$ are constructed.

Lemma 7.2. *At any time of the construction procedure, $l \leq r$. In particular, since $r \leq k$, we have $l_p \leq k$ for all periods $p \leq p_{max}$.*

Proof Fix a period p. We prove this by induction. The claim is satisfied for any $l = 1$ when $p \geq 2$ since in this case, at the first time $t = i_{p,1}$ of the period p we also construct the first vector $\boldsymbol{v}_{p,1}$. For $p = 1$, note that the first informative query t that falls in scenarios (2b) or (2c) is exploratory. Indeed, in these cases we have $F_{\boldsymbol{A},\boldsymbol{v},1,0}(\boldsymbol{x}_t) < \eta(-\gamma_1 - \gamma_2/2) \leq -\eta\gamma_1/2$, and the second criterion for an exploratory query is immediate $\|P_{Span(\boldsymbol{x}_{i_{1,r'}}, r' \leq 0)}(\boldsymbol{x}_t)\| = 0$ since no indices $i_{1,r}$ have been defined yet.

We now suppose that the claim holds for $l - 1 \geq 1$. Let $t_{p,l}$ be the time when $\boldsymbol{v}_{p,l}$ is constructed and $i_{p,1}, \ldots, i_{p,r}$ the indices constructed until the beginning of iteration $t_{p,l}$. If a new index $i_{p,r'}$ was constructed in times $(t_{p,l-1}, t_{p,l})$ then the claim holds immediately. Suppose that this is not the case. Note that $t_{p,l}$ falls in scenario (2b) which means in particular that

$$\eta(\boldsymbol{v}_{p,l-1}^\top \boldsymbol{x}_{t_{p,l}} - p\gamma_1 - (l-1)\gamma_2) \leq F_{\boldsymbol{A},\boldsymbol{v},p,l-1}(\boldsymbol{x}_{t_{p,l}}) < \eta(-p\gamma_1 - (l-1)\gamma_2 - \gamma_2/2).$$

As a result,

$$|\boldsymbol{y}_{p,l-1}^\top \boldsymbol{x}_{t_{p,l}}| \geq |\boldsymbol{v}_{p,l-1}^\top \boldsymbol{x}_{t_{p,l}}| - \delta > \frac{\gamma_2}{2} - \delta.$$

Next, when $r \geq l - 1$ is the number of indices constructed so far, we decompose $\boldsymbol{y}_{p,l-1} = \alpha_1 \boldsymbol{b}_{p,1} + \ldots + \alpha_r \boldsymbol{b}_{p,r} + \tilde{\boldsymbol{y}}_{p,l-1}$ where $\tilde{\boldsymbol{y}}_{p,l-1} \in Span(\boldsymbol{x}_{i_{p,r'}}, r' \leq r)^\perp$. Since by construction of $\boldsymbol{y}_{p,l-1}$ one has $|\alpha_{r'}| \leq d^{-3}$ for all $r' \leq r$, we have

$$\|\tilde{\boldsymbol{y}}_{p,l-1} - \boldsymbol{y}_{p,l-1}\| \leq \frac{\sqrt{r}}{d^3} \leq \frac{1}{d^2\sqrt{d}}.$$

Therefore,

$$\|P_{Span(\boldsymbol{x}_{i_{p,r'}}, r' \leq r)^\perp}(\boldsymbol{x}_{t_{p,l}})\| \geq |\tilde{\boldsymbol{y}}_{p,l-1}^\top \boldsymbol{x}_{t_{p,l}}| \geq |\boldsymbol{y}_{p,l-1}^\top \boldsymbol{x}_{t_{p,l}}| - \frac{1}{d^2\sqrt{d}} > \frac{\gamma_2}{2} - \frac{1}{d^2\sqrt{d}} - \delta \geq \frac{\gamma_2}{4}.$$

As a result, $t_{p,l}$ is exploratory, hence $i_{p,r+1} = t_{p,l}$. This ends the proof of the recursion and the lemma. ■

We recall that P and L denote the last defined period and vector $\boldsymbol{v}_{P,L}$. From Lemma 7.2, we have in particular $P \leq p_{max}$ and $L \leq k$. The next step involves showing that with high probability, the returned values and vectors returned by the above procedure are consistent with a first-order oracle for minimizing the function $F_{\boldsymbol{A},\boldsymbol{v},P,L}$, as stated in Proposition 7.1.

Proof of Proposition 7.1 Consider a given iteration t. We aim to show that we have $(f_t, \boldsymbol{g}_t) = (F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t), \partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t))$. By construction, if $t \geq d^2$, the result is immediate.

Now suppose $t \leq d^2$. We first consider the case when $\boldsymbol{x}_t$ is non-informative (1). By definition, $F_{\boldsymbol{A}}(\boldsymbol{x}_t) > \eta$. Since for any $(p,l) \leq_{lex} (P,L)$ one has $|\boldsymbol{v}_{p,l}^\top \boldsymbol{x}_t| \leq \|\boldsymbol{v}_{p,l}\|\|\boldsymbol{x}_t\| \leq 1$, we have

$$F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t) = \max\left\{F_{\boldsymbol{A}}(\boldsymbol{x}_t), \eta\left(\max_{(p,l)\leq_{lex}(P,L)} \boldsymbol{v}_{p,l}^\top \boldsymbol{x} - p\gamma_1 - l\gamma_2\right)\right\} = F_{\boldsymbol{A}}(\boldsymbol{x}_t).$$

As a result, the response of the procedure for $\boldsymbol{x}_t$ is consistent with $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ and the returned subgradient is $\tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t) = \partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t)$. Therefore, it suffices to focus on informative queries (2). We will denote by $t_{p,l}$ the index of the iteration when $\boldsymbol{v}_{p,l}$ has been defined, for $(p,l) \leq_{lex} (P,L)$. Consider a specific couple $(p,l) \leq_{lex} (P,L)$, and let r denote the number of constructed indices on or before $t_{p,l}$. Let $\boldsymbol{b}_{p,1}, \ldots, \boldsymbol{b}_{p,r}$ the corresponding vectors resulting from the Gram-Schmidt procedure on $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}}$. Then, conditionally on the history until time $t_{p,l}$, the vector $\boldsymbol{v}_{p,l}$ was defined as $\boldsymbol{v}_{p,l} = \phi_\delta(\boldsymbol{y}_{p,l})$, where $\boldsymbol{y}_{p,l}$ is sampled as $\sim \mathcal{U}(S^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p,r'}^\top \boldsymbol{z}| \leq d^{-3}, \forall r' \leq r\})$. As a result, from Lemma 7.5, for any $t \leq t_{p,l}$, we have

$$\mathbb{P}\left(|\boldsymbol{x}_t^\top \boldsymbol{v}_{p,l}| \geq 3\sqrt{\frac{2\log d}{d}} + \frac{2}{d^2}\right) \leq \frac{6\sqrt{2\log d}}{d^6}.$$

We then define the following event

$$\mathcal{E} = \bigcap_{(p,l)\leq_{lex}(P,L)} \bigcap_{t\leq t_{p,l}} \left\{|\boldsymbol{x}_t^\top \boldsymbol{v}_{p,l}| < 3\sqrt{\frac{2\log d}{d}} + \frac{2}{d^2}\right\},$$

which by the union bound has probability $\mathbb{P}(\mathcal{E}) \geq 1 - 3\sqrt{2\log d}/d^2$. We are now ready to show that the construction procedure is consistent with optimizing $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ on the event $\mathcal{E}$. As seen before, we can suppose that $\boldsymbol{x}_t$ is informative (2). Using the same notations as before, because $\mathcal{E}$ is met, for any $p < p' \leq P$ and $l' \leq l_{p'}$, we have for $d \geq 2$,

$$\boldsymbol{v}_{p',l'}^\top \boldsymbol{x}_t - p'\gamma_1 - l'\gamma_2 < 3\sqrt{\frac{2\log d}{d}} + \frac{1}{d} - p\gamma_1 - \gamma_1 \leq -p\gamma_1 - \frac{\gamma_1}{2} \leq -p\gamma_1 - d\gamma_2 - \frac{\gamma_2}{2},$$

where we used $3\sqrt{2} + 1 \leq 6$ and $2d\gamma_2 \leq \gamma_1/2$. As a result, we obtain that

$$\max_{(p',l')\leq_{lex}(P,L), p'>p} \boldsymbol{v}_{p',l'}^\top \boldsymbol{x}_t - p'\gamma_1 - l'\gamma_2 < -p\gamma_1 - l\gamma_2 - \frac{\gamma_2}{2}.$$

Next, we consider the case of vectors $\boldsymbol{v}_{p,l'}$ where $l \leq l' \leq l_p$ and $t_{p,l'} \geq t$ (this also includes the case when we defined $\boldsymbol{v}_{p,l}$ at time $t = t_{p,l}$). We write $\tilde{l}$ for the smallest such index l. As a remark, $\tilde{l} \in \{l, l+1\}$. Note that if such indices exist, this means that before starting iteration t, the procedure has not yet reached $r = k$. There are two cases. If $\boldsymbol{x}_t$ was exploratory, we have $t = i_{p,r}$ hence $\|P_{Span(\boldsymbol{b}_{p,r'}, r'\leq r)^\top}(\boldsymbol{x}_t)\| = 0$. If $\boldsymbol{x}_t$ is not exploratory, either

$$\|P_{Span(\boldsymbol{b}_{p,r'}, r'\leq r)^\top}(\boldsymbol{x}_t)\| < \frac{\gamma_2}{4}\|\boldsymbol{x}_t\| \leq \frac{\gamma_2}{4}, \tag{7.4}$$

or we have $F_{\boldsymbol{A},v,p,l}(\boldsymbol{x}_t) > -\eta\gamma_1/2$. We start with the last scenario when $F_{\boldsymbol{A},v,p,l}(\boldsymbol{x}_t) > -\eta\gamma_1/2$. Then, on $\mathcal{E}$, one has

$$\max_{(p,l)<_{lex}(p',l')\leq_{lex}(P,L)} \boldsymbol{v}_{p',l'}^\top \boldsymbol{x}_t - p'\gamma_1 - l'\gamma_2 \leq -\gamma_1 + 3\sqrt{\frac{2\log d}{d}} + \frac{1}{d} \leq \frac{\gamma_1}{2}$$

As a result, this shows that $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t) = F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t)$. Hence using a first-order oracle from $F_{\boldsymbol{A},\boldsymbol{v},l,p}$ at $\boldsymbol{x}_t$ is already consistent with $F_{\boldsymbol{A},\boldsymbol{v},P,L}$. Thus, for whichever step (2a), (2b) or (2c) is performed, since these can only increase the knowledge on $\boldsymbol{v}$, the response given by the construction procedure is consistent with minimizing $F_{\boldsymbol{A},\boldsymbol{v}}$.

It remains to treat the first two scenarios in which we always have Eq (7.4). In particular, when writing $\boldsymbol{x}_t = \alpha_1 \boldsymbol{b}_{p,1} + \ldots + \alpha_r \boldsymbol{b}_{p,r} + \tilde{\boldsymbol{x}}_t$ where $\tilde{\boldsymbol{x}}_t = P_{Span(\boldsymbol{b}_{p,r'}, r' \leq r)^\perp}(\boldsymbol{x}_t)$, we have $\|\tilde{\boldsymbol{x}}_t\| < \frac{\gamma_2}{4}$. As a result, for $\tilde{l} \leq l' \leq l_p$, one has for

$$\begin{aligned}
|\boldsymbol{v}_{p,l'}^\top \boldsymbol{x}_t| \leq |\boldsymbol{y}_{p,l'}^\top \boldsymbol{x}_t| + \delta &\leq |\alpha_1||\boldsymbol{y}_{p,l'}^\top \boldsymbol{b}_{p,1}| + \ldots + |\alpha_r||\boldsymbol{y}_{p,l'}^\top \boldsymbol{b}_{p,r}| + \|\tilde{\boldsymbol{x}}_t\| + \delta \\
&< \|\boldsymbol{\alpha}\|_1 \frac{1}{d^3} + \frac{\gamma_2}{4} + \delta \\
&\leq \frac{\gamma_2}{4} + \frac{1}{d^2\sqrt{d}} + \frac{1}{d^3} \leq \frac{\gamma_2}{2},
\end{aligned}$$

where in the last inequality we used $d \geq 3$. As a result, provided that $\tilde{l}$ exists, this shows that

$$\max_{\tilde{l} \leq l' \leq l_p} \boldsymbol{v}_{p,l'}^\top \boldsymbol{x}_t - p\gamma_1 - l'\gamma_2 = \boldsymbol{v}_{p,\tilde{l}}^\top \boldsymbol{x}_t - p\gamma_1 - \tilde{l}\gamma_2 < -p\gamma_1 - \tilde{l}\gamma_2 + \frac{\gamma_2}{2}. \tag{7.5}$$

On the other hand, if $t = i_{p+1,1}$, the same reasoning works for t viewing it as in period $p+1$, which shows for this case that

$$\max_{l' \leq l_{p+1}} \boldsymbol{v}_{p+1,l'}^\top \boldsymbol{x}_t - (p+1)\gamma_1 - l'\gamma_2 = \boldsymbol{v}_{p+1,1}^\top \boldsymbol{x}_t - (p+1)\gamma_1 - \gamma_2 < -(p+1)\gamma_1 - \frac{\gamma_2}{2}. \tag{7.6}$$

As a conclusion of these estimates, we showed that on $\mathcal{E}$, we have

$$F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t) = \max\left\{F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t), \eta(\boldsymbol{v}_{p',l'}^\top \boldsymbol{x}_t - p'\gamma_1 - l'\gamma_2)\right\} := \tilde{F}_{\boldsymbol{A},\boldsymbol{v},t}(\boldsymbol{x}_t)$$

where (p', l') is the very next vector that is defined after starting iteration t (potentially, it has $t_{p',l'} = t$ if we defined a vector at this time). It then suffices to check that the value and vector returned by the procedure are consistent with the right-hand side. By construction, if we constructed $\boldsymbol{v}_{p',l'}$ at step t: case (2b) or (2c), then the procedure directly uses a first-order oracle for $\tilde{F}_{\boldsymbol{A},\boldsymbol{v},t}$. Further, by construction of the subgradients since they break ties lexicographically in (p, l), the returned subgradient is exactly $\partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t)$. It remains to check that this is the case when no vector $\boldsymbol{v}_{p',l'}$ is defined at step t: case (2a). This corresponds to the case when $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t) \geq \eta(-p\gamma_1 - l\gamma_2 - \gamma/2)$. In this case, the upper bound estimates from Eq (7.5) and Eq (7.6) imply that

$$\boldsymbol{v}_{p',l'}^\top \boldsymbol{x}_t - p'\gamma_1 - l'\gamma_2 < -p\gamma_1 - l\gamma_2 - \gamma/2,$$

and as a result, $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t) = F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_t)$. Therefore, using a first-order oracle of $F_{\boldsymbol{A},\boldsymbol{v},p,l}$ at $\boldsymbol{x}_t$ is valid, and the break of ties of the subgradient of $\tilde{F}_{\boldsymbol{A},\boldsymbol{v},t}$ is the same as the break of ties of $\partial F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t)$. This ends the proof that on $\mathcal{E}$ the procedure gives responses consistent with an optimization oracle for $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ with subgradient function $\partial F_{\boldsymbol{A},\boldsymbol{v},P,L}$. Because $\mathbb{P}(\mathcal{E}) \geq 1 - C\sqrt{\log d}/d^2$ for some constant $C > 0$, this ends the proof of the proposition. ■

Last, we provide an upper bound on the optimal value of $F_{\boldsymbol{A},\boldsymbol{v},P,L}$.

Proposition 7.5. *Let $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n\times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$. For any algorithm alg for convex optimization, let $\boldsymbol{v}$ be the resulting set of vectors constructed by the randomized procedure. With probability at least $1 - C\sqrt{\log d}/d$ over the randomness of $\boldsymbol{A}$, $\boldsymbol{v}_0$ and $\boldsymbol{v}$, we have*

$$\min_{\boldsymbol{x}\in B_d(\boldsymbol{0},1)} F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) \le -\frac{\eta}{30\sqrt{(kp_{max}+1)\log d}},$$

for some universal constant $C > 0$.

Proof For simplicity, let us enumerate all the constructed vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{l_{max}}$ by order of construction. Hence, $l_{max} \le p_{max}k$. We use the same enumeration for $\boldsymbol{y}_1, \dots, \boldsymbol{y}_{l_{max}}$. Next, let $C_d = \sqrt{40(l_{max}+1)\log d}$ and consider the following vector,

$$\bar{\boldsymbol{x}} = -\frac{1}{C_d}\sum_{l=0}^{l_{max}} P_{Span(\boldsymbol{a}_i, i\le n)^\perp}(\boldsymbol{v}_l).$$

In particular, note that we included $\boldsymbol{v}_0$ in the sum. For convenience, we write $P_{\boldsymbol{A}^\perp}$ instead of $P_{Span(\boldsymbol{a}_i,i\le n)^\perp}$. Also, for convenience let us define $\boldsymbol{z}_l = \sum_{l'\le l} P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)$. Fix an index $1 \le l \le l_{max}$. Then, by Lemma 7.5, with $t_0 := \sqrt{\frac{6\log d}{d}} + \frac{2}{d^2}$, we have

$$\begin{aligned}
\mathbb{P}\left(|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{l+1})^\top \boldsymbol{z}_l| > t_0\|\boldsymbol{z}_l\|\right) &= \mathbb{P}\left(|\boldsymbol{v}_{l+1}^\top P_{\boldsymbol{A}^\perp}(\boldsymbol{z}_l)| > t_0\|\boldsymbol{z}_l\|\right)\\
&\le \mathbb{P}\left(|\boldsymbol{v}_{l+1}^\top P_{\boldsymbol{A}^\perp}(\boldsymbol{z}_l)| > t_0\|P_{\boldsymbol{A}^\perp}(\boldsymbol{z}_l)\|\right)\\
&\le \frac{2\sqrt{6\log d}}{d^2}.
\end{aligned}$$

Similarly, we have that

$$\mathbb{P}\left(|\boldsymbol{v}_{l+1}^\top \boldsymbol{z}_l| > t_0\|\boldsymbol{z}_l\|\right) \le \frac{2\sqrt{6\log d}}{d^2}.$$

We consider the event $\mathcal{E} = \bigcap_{l\le l_{max}}\{|\boldsymbol{v}_l^\top \boldsymbol{z}_{l-1}|, |P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)^\top \boldsymbol{z}_{l-1}| \le t_0\|\boldsymbol{z}_l\|\}$, which since $l_{max} \le d$, by the union bound has probability at least $1 - 4\sqrt{6\log d}/d$. Then, on $\mathcal{E}$, for any $l < l_{max}$,

$$\|\boldsymbol{z}_{l+1}\|^2 \le \|\boldsymbol{z}_l\|^2 + \|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{l+1})\|^2 + 2|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{l+1})^\top \boldsymbol{z}_l| \le \|\boldsymbol{z}_l\|^2 + 1 + 2t_0\|\boldsymbol{z}_l\|.$$

We now prove by induction that $\|\boldsymbol{z}_l\|^2 \le 40\log d\cdot(l+1)$, which is clearly true for $\boldsymbol{z}_0$ since $\|\boldsymbol{z}_0\| = \|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_0)\| \le \|\boldsymbol{v}_0\| \le 1$. Suppose this is true for $l < l_{max}$. Then, using the above equation and the fact that $t_0 \le 3\sqrt{\frac{\log d}{d}}$ for $d \ge 4$,

$$\|\boldsymbol{z}_{l+1}\|^2 \le 40\log d\cdot(l+1) + 1 + 6\sqrt{40}\log d\sqrt{\frac{l+1}{d}} \le 40\log d\cdot(l+2),$$

where we used $l_{max}+1 \le d$, which completes the induction. In particular, on $\mathcal{E}$, we have that $\|\bar{\boldsymbol{x}}\| \le 1$. Also, observe that by construction $\bar{\boldsymbol{x}} \in Span(\boldsymbol{a}_i, i\le n)^\perp$ so that $\|\boldsymbol{A}\bar{\boldsymbol{x}}\|_\infty = 0$. Next, for any $0 \le l \le l_{max}$, we have

$$\boldsymbol{v}_l^\top \bar{\boldsymbol{x}} = -\frac{\boldsymbol{v}_l^\top \boldsymbol{z}_{l_{max}}}{C_d} = -\frac{1}{C_d}\left(\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\|^2 + \boldsymbol{v}_l^\top \boldsymbol{z}_{l-1} + \sum_{l<l'\le l_{max}} \boldsymbol{v}_l^\top P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{l'})\right).$$

We will give estimates on each term of the above equation. First, if the indices $i_{p,1}, \ldots, i_{p,r}$ were defined before defining $\boldsymbol{v}_l$, we denote $\tilde{\boldsymbol{y}} = P_{Span(\boldsymbol{x}_{i_{p,r'}}, r' \leq r)^\perp}(\boldsymbol{y}_l)$, the component of $\boldsymbol{y}_l$ which is perpendicular to the explored space at that time. Then, we can write $\boldsymbol{y}_l = \alpha_1^l \boldsymbol{b}_{p,1} + \ldots + \alpha_r^l \boldsymbol{b}_{p,1} + \tilde{\boldsymbol{y}}_l$, and note that

$$\|\tilde{\boldsymbol{y}}_l\| = \sqrt{\|\boldsymbol{y}_l\| - (\alpha_1^l)^2 - \ldots - (\alpha_r^l)^2} \geq \sqrt{1 - \frac{k}{d^6}} \geq 1 - \frac{1}{d^5}.$$

Then, we have

$$\begin{aligned}
\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\| &\geq \|P_{\boldsymbol{A}^\perp}(\boldsymbol{y}_l)\| - \delta \\
&\geq \|P_{Span(\boldsymbol{a}_i, i \leq n, \boldsymbol{b}_{p,r'}, r \leq r')^\perp}(\boldsymbol{y}_l)\| - \delta \\
&= \|P_{Span(\boldsymbol{a}_i, i \leq n, \boldsymbol{b}_{p,r'}, r \leq r')^\perp}(\tilde{\boldsymbol{y}}_l)\| - \delta \\
&\geq \left\| P_{Span(a_i, i \leq n, \boldsymbol{b}_{p,r'}, r' \leq r)^\perp}\left(\frac{\tilde{\boldsymbol{y}}_l}{\|\tilde{\boldsymbol{y}}_l\|}\right)\right\| - \frac{1}{d^5} - \delta.
\end{aligned}$$

As a result, since $\delta = d^{-3}$, this shows that

$$\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\|^2 \geq \left\| P_{Span(a_i, i \leq n, \boldsymbol{b}_{p,r'}, r' \leq r)^\perp}\left(\frac{\tilde{\boldsymbol{y}}_l}{\|\tilde{\boldsymbol{y}}_l\|}\right)\right\|^2 - 2\delta.$$

Now observe that $dim(Span(a_i, i \leq n, \boldsymbol{b}_{p,r'}, r' \leq r)^\perp) \geq d - n - k$, while $\frac{\tilde{\boldsymbol{y}}_l}{\|\tilde{\boldsymbol{y}}_l\|}$ is a uniformly random unit vector in $Span(\boldsymbol{b}_{p,r'}, r \leq r')^\perp$. Therefore, using Proposition 7.10 we obtain for $t < 1$,

$$\begin{aligned}
&\mathbb{P}\left(\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\|^2 + 2\delta - \frac{d-n-k}{d} \leq -t\right) \\
&\leq \mathbb{P}\left(\left\| P_{Span(a_i, i \leq n, \boldsymbol{b}_{p,r'}, r' \leq r)^\perp}\left(\frac{\tilde{\boldsymbol{y}}_l}{\|\tilde{\boldsymbol{y}}_l\|}\right)\right\|^2 - \frac{d-n-k}{d} \leq -t\right) \\
&\leq e^{-(d-k)t^2}.
\end{aligned}$$

As a result since $d - n - k \geq d/2$, we obtain

$$\mathbb{P}\left(\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\|^2 \leq \frac{1}{2} - 2\sqrt{\frac{\log d}{d}} - 2\delta\right) \leq \frac{1}{d^2}.$$

Nwxt, define $\mathcal{F} = \bigcap_{l \leq l_{max}} \{\|P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_l)\|^2 \geq \frac{1}{2} - 2\sqrt{\frac{\log d}{d}} - 2\delta\}$, which since $l_{max} + 1 \leq d$ and by the union bound has probability at least $\mathbb{P}(\mathcal{F}) \geq 1 - 1/d$. Next, we turn to the last term. For any $0 \leq l < l_{max}$, we focus on the sequence $(\sum_{l'=l+1}^{l+u} \boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{y}_{l'}))_{1 \leq u \leq l_{max}-l}$ and first note that this is a martingale. These increments are symmetric (because $\boldsymbol{y}_{l'}$ is symmetric) even conditionally on $\boldsymbol{A}$ and $\boldsymbol{v}_l, \boldsymbol{y}_l, \ldots, \boldsymbol{y}_{l'-1}$. Next, let $t_1 = 2\sqrt{\frac{3\log d}{d}} + \frac{2}{d^2}$. Note that for $d \geq 4$, we have $t_1 \leq 4\sqrt{\frac{\log d}{d}}$. Further, by Lemma 7.5,

$$\mathbb{P}(|\boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{y}_{l'})| > t_1) = \mathbb{P}(|P_{\boldsymbol{A}^\top}(\boldsymbol{v}_l)^\top \boldsymbol{y}_{l'}| > t_1) \leq \frac{4\sqrt{3\log d}}{d^4},$$

where we used the fact that $P_{\boldsymbol{A}^\perp}$ is a projection. Let $\mathcal{G}_l = \bigcap_{l<l'\leq l_{max}} \{|\boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{v}_{l'})| \leq t_1\}$, which by the union bound has probability $\mathbb{P}(\mathcal{G}_l) \geq 1 - 4\sqrt{3\log d}/d^3$. Next, we define $I_{l,u} = (\boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{y}_{l+u}) \wedge t_1) \vee (-t_1)$, the increments capped at absolute value t_1. Because $\boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{y}_{l+u})$ is symmetric, so is $I_{l,u}$. As a result, these are bounded increments of a martingale, to which we can apply the Azuma-Hoeffding inequality.

$$\mathbb{P}\left(\left|\sum_{u=1}^{l_{max}-l} I_{l,u}\right| \leq 2t_1\sqrt{(l_{max}-l)\log d}\right) \geq 1 - \frac{2}{d^2}.$$

We denote by $\mathcal{H}_l$ this event. Observe that on $\mathcal{G}_l$, the increments $I_{l,u}$ and $\boldsymbol{v}_l^\top P_{\boldsymbol{A}^\top}(\boldsymbol{y}_{l+u})$ coincide for all $1 \leq u \leq l_{max} - l$. As a result, on $\mathcal{G}_l \cap \mathcal{H}_l$ we obtain

$$\begin{aligned}
\left|\sum_{l<l'\leq l_{max}} \boldsymbol{v}_l^\top P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{l'})\right| &\leq \left|\sum_{l<l'\leq l_{max}} \boldsymbol{v}_l^\top P_{\boldsymbol{A}^\perp}(\boldsymbol{y}_{l'})\right| + (l_{max}-1)\delta \\
&\leq \left|\sum_{u=1}^{l_{max}-l} I_{l,u}\right| + (d-2)\delta \\
&\leq 2t_1\sqrt{l_{max}\log d} + (d-2)\delta.
\end{aligned}$$

Then, on the event $\mathcal{E} \cap \mathcal{F} \cap \bigcap_{l\leq l_{max}} \mathcal{G}_l \cap \mathcal{H}_l$, for any $1 \leq l \leq l_{max}$ one has

$$\begin{aligned}
\boldsymbol{v}_l^\top \boldsymbol{z}_{l_{max}} &\geq \frac{1}{2} - 2\sqrt{\frac{\log d}{d}} - t_0\|\boldsymbol{z}_l\| - 2t_1\sqrt{l_{max}\log d} - \frac{1}{d^2} \\
&\geq \frac{1}{2} - 2\sqrt{\frac{\log d}{d}} - 3\log d\sqrt{40\frac{l_{max}+1}{d}} - 8\log d\sqrt{\frac{l_{max}}{d}} - \frac{1}{d^2} \\
&\geq \frac{1}{2} - 30\log d\sqrt{\frac{l_{max}+1}{d}} \\
&\geq \frac{1}{6},
\end{aligned}$$

where in the last inequalities we used the fact that $l_{max} \leq kp_{max} \leq c_{d,1}d - 1$ where $c_{d,1} = \frac{1}{90^2\log^2 d}$ as per Eq (7.3). As a result, we obtain that on $\mathcal{E} \cap \mathcal{F} \cap \bigcap_{l\leq l_{max}} \mathcal{G}_l \cap \mathcal{H}_l$, which has probability at most $1 - C\sqrt{\log d}/d$ for some constant $C > 0$,

$$\max_{p\leq p_{max}, l\leq k} \boldsymbol{v}_{p,l}^\top \bar{\boldsymbol{x}} \leq -\frac{1}{6C_d} \leq -\frac{1}{40\sqrt{(kp_{max}+1)\log d}}.$$

Since $\|\boldsymbol{A}\bar{\boldsymbol{x}}\|_\infty = 0$, and $\eta \geq \frac{\eta}{40\sqrt{(kp_{max}+1)\log d}}$, this shows that

$$F_{\boldsymbol{A},\boldsymbol{v}}(\bar{\boldsymbol{x}}) \leq -\frac{\eta}{40\sqrt{(kp_{max}+1)\log d}}.$$

This ends the proof of the proposition. ■

7.3.4 Reduction from convex optimization to the optimization procedure

Next, we prove Proposition 7.2 which shows that to be successful for the optimization procedure, an algorithm needs to properly define the function $F_{\boldsymbol{A},v}$, i.e., to complete all periods until p_{max}.

Proof of Proposition 7.2 Let $\boldsymbol{x}^\star(alg) = \boldsymbol{x}_T$ denote the final answer of alg when run with the optimization procedure. By hypothesis, we have $T \leq d^2$. As before, let $P \leq p_{max}$ and $L \leq k$ be the indices such that the last vector constructed by the optimization procedure is $v_{P,L}$. Let $\mathcal{E}$ be the event when alg run on the optimization procedure does not end period p_{max}. We focus on $\mathcal{E}$ and consider two cases.

First, suppose that $T > t_{P,L}$, i.e., the last vector was not constructed at time T. As a result, this means that $\boldsymbol{x}_T$ corresponds either to a non-informative query—scenario (1)—in which case $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_T) \geq F_{\boldsymbol{A}}(\boldsymbol{x}_T) \geq \eta$, or this means that $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_t) \geq \eta(-P\gamma_1 - L\gamma_2 - \gamma/2)$—scenario (2a).

Second, we suppose that $T = t_{P,L}$, i.e., the last vector was constructed at time T. Then, by construction of $\boldsymbol{v}_{P,L}$ and $\boldsymbol{y}_{P,L}$, we have indices $i_{P,1}, \ldots, i_{P,r} \leq T$ such that with the Gram-Schmidt decomposition $\boldsymbol{b}_{P,1}, \ldots, \boldsymbol{b}_{P,r}$ of $\boldsymbol{x}_{i_{P,1}}, \ldots, \boldsymbol{x}_{i_{P,r}}$, we have $|\boldsymbol{b}_{p,r'}^\top \boldsymbol{y}_{P,L}| \leq d^{-3}$ for all $r' \leq r$. In particular, writing $\boldsymbol{x}_T = \alpha_1 \boldsymbol{b}_{P,1} + \ldots + \alpha_r \boldsymbol{b}_{P,r} + \tilde{\boldsymbol{x}}_T$, where $\tilde{\boldsymbol{x}}_T \in Span(\boldsymbol{x}_{i_{P,r'}}, r' \leq r)^\perp$, either we have $i_{P,r} = T$, in which case $\tilde{\boldsymbol{x}}_T = \boldsymbol{0}$, or $\boldsymbol{x}_T$ was not exploratory in which case we directly have $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_T) \geq F_{\boldsymbol{A},\boldsymbol{v},P,L-1}(\boldsymbol{x}_T) > -\eta\gamma_1/2$, or we have $\|\tilde{\boldsymbol{x}}_T\| < \|\boldsymbol{x}_T\|\gamma_2/4 \leq \gamma_2/4$. For all remaining cases to consider, we obtain

$$|\boldsymbol{v}_{P,L}^\top \boldsymbol{x}_T| \leq |\boldsymbol{y}_{P,L}^\top \boldsymbol{x}_T| + \delta \leq \frac{\|\boldsymbol{\alpha}\|_1}{d^3} + \|\tilde{\boldsymbol{x}}_T\| + \delta \leq \frac{1}{d^3} + \frac{1}{d^2\sqrt{d}} + \frac{\gamma_2}{4} < \frac{\gamma_2}{2}.$$

In the last inequality, we used $d \geq 4$. This shows that $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}_T) \geq \eta(-P\gamma_1 - L\gamma_2 - \gamma_2/2)$. As a result, in all cases this shows that $F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}^\star(alg)) \geq \eta(-P\gamma_1 - L\gamma_2 - \gamma_2/2) \geq -\eta(p_{max}+1)\gamma_1$. Now define the event

$$\mathcal{F} = \left\{ \min_{\boldsymbol{x} \in B_d(\boldsymbol{0},1)} F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) \leq -\frac{\eta}{40\sqrt{(kp_{max}+1)\log d}} \right\}.$$

By Proposition 7.5 we have $\mathcal{P}(\mathcal{F}) \geq 1 - C\sqrt{\log d}/d$. From Eq (7.3),

$$(p_{max}+1)^{3/2} \leq \frac{1}{60\gamma_1\sqrt{k\log d}}.$$

Thus,

$$(p_{max}+1)\gamma_1 \leq \frac{1}{60\sqrt{k(p_{max}+1)\log d}} \leq \frac{1}{60\sqrt{(kp_{max}+1)\log d}}$$

Then, since $F_{\boldsymbol{A},\boldsymbol{v},P,L} \leq F_{\boldsymbol{A},\boldsymbol{v}}$, this shows that on $\mathcal{E} \cap \mathcal{F}$,

$$\begin{aligned} F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}^\star(alg)) \geq -\eta(p_{max}+1)\gamma_1 &\geq \min_{\boldsymbol{x} \in B_d(\boldsymbol{0},1)} F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) + \frac{\eta}{120\sqrt{(kp_{max}+1)\log d}} \\ &> \min_{\boldsymbol{x} \in B_d(\boldsymbol{0},1)} F_{\boldsymbol{A},\boldsymbol{v},P,L}(\boldsymbol{x}) + \frac{\eta}{2\sqrt{d}} \end{aligned}$$

where in the last inequality, we used $kp_{max} \leq c_{d,1}d-1$. As a result, letting $\mathcal{G}$ be the event when alg succeeds for precision $\epsilon = \eta/(2\sqrt{d})$. By hypothesis, $\mathcal{P}(\mathcal{G}) \geq q$. By the above equations, one has $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G} = \emptyset$. Therefore, $\mathbb{P}(\mathcal{G} \cap \mathcal{E}^c) \geq \mathcal{P}(\mathcal{G}) - \mathbb{P}(\mathcal{G} \cap \mathcal{E} \cap \mathcal{F}) - \mathbb{P}(\mathcal{F}^c) \geq q - C\sqrt{\log d}/d$. This ends the proof of the proposition. ■

7.3.5 Reduction of the optimization procedure to the Orthogonal Vector Game with Hints

Using the result from Proposition 7.2, we show that solving the optimization procedure implies solving the Orthogonal Game with Hints with high probability.

Proof of Proposition 7.3 Let alg be an M-bit algorithm solving the feasibility problem with mp_{max} queries with probability at least q. Below, we describe the strategy for Game 7.2.

In the first part of the strategy, the player observes $\boldsymbol{A}$. First, submit an empty query to the Oracle to obtain a vector $\boldsymbol{v}_0$, which as a result is uniformly distributed among $\mathcal{D}_\delta$. We then proceed to simulate the optimization procedure for alg using parameters $\boldsymbol{A}$ and $\boldsymbol{v}_0$ (lines 3-6 of Game 7.2). Precisely, whenever a new vector $\boldsymbol{v}_{p,l}$ needs to be defined according to the optimization procedure, the player submits the corresponding vectors $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,r}}$ to the oracle and receives in return a vector which defines $\boldsymbol{v}_{p,l}$. In this manner, the player simulates exactly the optimization procedure. In all cases, the number of queries in this first phase is at most $1 + kp_{max} \leq d$. For the remaining queries to perform, the player can query whichever vectors, these will not be used in the rest of the strategy. If the simulation did not end period p_{max}, the complete procedure fails. We now describe the rest of the procedure when period p_{max} was ended. During the simulation, the algorithm records the time $i_{p,1}$ when period p started for all $p \leq p_{max}+1$. Recall that for $p_{max}+1$, we only define $i_{p_{max}+1,1}$, this is the time that ends period p_{max}. By hypothesis, $i_{p_{max}+1,1} \leq mp_{max}$. As a result, there must be a period $p \leq p_{max}$ which uses at most m queries: $i_{p+1,1} - i_{p,1} \leq m$. We define the memory Message to be the memory of alg just before starting iteration $i_{p,1}$, at the beginning of period p (line 7 of Game 7.2). Next, since the period p_{max} was ended, the vectors $\boldsymbol{v}_{p,l}$ for $p \leq p_{max}, l \leq l_p$ were all defined. The player can therefore submit the function $\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}}$ to the Oracle (line 8 of Game 7.2) as follows,

$$\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}} : \boldsymbol{x} \mapsto \begin{cases} (\boldsymbol{g}_{\boldsymbol{A}}(\boldsymbol{x}), 1) & \text{if } F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \\ (\boldsymbol{v}_0, 2) & \text{otherwise and if } F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}) = \eta \boldsymbol{v}_0^\top \boldsymbol{x}, \\ (\boldsymbol{v}_{p,l}, 2 + (p-1)k + l) & \text{otherwise and if} \\ & (p,l) = \underset{(p',l') \leq_{lex} (p_{max}, l_{p_{max}})}{\arg\max} \boldsymbol{v}_{p',l'}^\top \boldsymbol{x} - p\gamma_1 - l\gamma_2. \end{cases} \tag{7.7}$$

Intuitively, the first component of $\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}}$ gives the subgradient $\partial F_{\boldsymbol{A},\boldsymbol{v}}$ to the following two exceptions: we always return $\boldsymbol{a}_i$ instead of $\pm\boldsymbol{a}_i$ and we return $\boldsymbol{v}_0$ (resp. $\boldsymbol{v}_{p,l}$) instead of $\eta\boldsymbol{v}_0$ (resp. $\eta\boldsymbol{v}_{p,l}$). The second term of $\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}}$ has values in $[2+p_{max}k]$. Hence, since $2+p_{max}k \leq d^2$, the function $\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}}$ takes values in $(\{\boldsymbol{a}_j, j \leq n\} \cup \{\boldsymbol{v}_l, l \leq d\}) \times [d^2]$.

The strategy then proceeds to play the Orthogonal Vector Game in a second part (lines 9-12 of Game 7.2) and use the responses of the Oracle to simulate the run of alg for the

optimization procedure in period p. To do so, we set the memory state of the algorithm alg to be Message. Then, for the next m iterations we proceed as follows. At iteration i of the process, we run alg with its current state to obtain a new query $\boldsymbol{z}_i$ which is then submitted to the oracle of the Orthogonal Vector Game, to get a response $(\boldsymbol{g}_i, s_i)$. We then use this response to simulate the response that was given by the optimization procedure in the first phase, computing $(v_i, \tilde{\boldsymbol{g}}_i)$ as follows

$$(v_i, \tilde{\boldsymbol{g}}_i) = \begin{cases} (|\boldsymbol{g}_i^\top \boldsymbol{z}_i| - \eta, sign(\boldsymbol{g}_i^\top \boldsymbol{z}_i)\boldsymbol{g}_i) & s_i = 1, \\ (\eta \boldsymbol{g}_i^\top \boldsymbol{z}_i, \eta \boldsymbol{g}_i) & s_i = 2, \\ (\eta(\boldsymbol{g}_i^\top \boldsymbol{z}_i - p\gamma_1 - l\gamma_2), \eta \boldsymbol{g}_i) & s_i = 2 + (p-1)k + l, p \leq p_{max}, 1 \leq l \leq k. \end{cases} \tag{7.8}$$

We can easily check that in all cases, $v_i = F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{z}_i)$ and that $\tilde{\boldsymbol{g}}_i = \partial F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{z}_i)$. We then pass $(\boldsymbol{v}_i, \tilde{\boldsymbol{g}}_i)$ as response to alg for the query $\boldsymbol{z}_i$ so it can update its state. Further, having defined $i_1 = 1$, the player can keep track of exploratory queries by checking whether

$$v_i \leq -\frac{\eta\gamma_1}{2} \quad \text{and} \quad \frac{\|P_{Span(\boldsymbol{z}_{i_{r'}}, r' \leq r)^\perp}(\boldsymbol{z}_i)\|}{\|\boldsymbol{z}_i\|} \geq \frac{\gamma_2}{4},$$

where $i_1, \ldots, i_r$ are the indices defined so far. We perform m such iterations unless alg stops and use the last remaining queries arbitrarily. Next, we check if the last index i_k was defined. If not, we pose $i_k = m+1$ and let $\boldsymbol{z}_{m+1}$ be the next query of alg. The final returned vectors are $\frac{\boldsymbol{z}_{i_1}}{\|\boldsymbol{z}_{i_1}\|}, \ldots, \frac{\boldsymbol{z}_{i_k}}{\|\boldsymbol{z}_{i_k}\|}$. This ends the description of the player's strategy.

We now show that the player wins with good probability. First, since alg makes at most $mp_{max} \leq d^2$ queries, by Proposition 7.2, on an event $\mathcal{E}$ of probability at least $q - C\sqrt{\log d}/d$, alg succeeds and ends the period p_{max}. On $\mathcal{E}$, by construction, the first phase of the strategy does not fail. Next, we show that in the second phase (lines 9-12 of Game 7.2), the queried vectors coincide exactly with the queried vectors from the corresponding period p in the first phase (lines 3-6 of Game 7.2). To do so, we only need to check that the responses provided to alg coincide with the response given by the optimization procedure. First, recall that on $\mathcal{E}$, all periods are completed, hence $F_{\boldsymbol{A},\boldsymbol{v},P,L} = F_{\boldsymbol{A},\boldsymbol{v}}$. Next, by Proposition 7.1, the responses of the procedure are consistent with optimizing $F_{\boldsymbol{A},\boldsymbol{v},P,L}$ and subgradients $\partial F_{\boldsymbol{A},\boldsymbol{v},P,L}$ on an event $\mathcal{F}$ of probability at least $1 - C'\sqrt{\log d}/d^2$. Therefore, on $\mathcal{E} \cap \mathcal{F}$, it suffices to check that the responses provided to alg are consistent with $F_{\boldsymbol{A},\boldsymbol{v}}$, which we already noted: at every step i, $(v_i, \tilde{\boldsymbol{g}}_i) = (F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{z}_i), \partial F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{z}_i))$. This proves that the responses and queries coincide exactly with those given by the optimization procedure on $\mathcal{E} \cap \mathcal{F}$.

Next, by construction, the chosen phase p had at most m iterations. Thus, on $\mathcal{E} \cap \mathcal{F}$, among $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m+1}$, we have the vectors $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,k}}$. Further, if i_k was not defined during part 2 of the strategy, this means that $i_k = m+1$, as defined in the player's strategy (line 21-22 of Algorithm 7.3). As a result, for all $u \leq k$, we have $\boldsymbol{z}_{i_u} = \boldsymbol{x}_{i_{p,u}}$. We now show that the returned vectors $\frac{\boldsymbol{x}_{i_{p,1}}}{\|\boldsymbol{x}_{i_{p,1}}\|}, \ldots, \frac{\boldsymbol{x}_{i_{p,k}}}{\|\boldsymbol{x}_{i_{p,k}}\|}$ are successful for Game 7.2. First, because $i_{p,1}, \ldots, i_{p,k}$ are exploratory queries, we have directly for $u \leq k$,

$$\frac{\|P_{Span(\boldsymbol{x}_{i_{p,v}}, v<u)^\perp}(\boldsymbol{x}_{i_{p,u}})\|}{\|\boldsymbol{x}_{i_{p,u}}\|} \geq \frac{\gamma_2}{4}.$$

Input: d, k, p_{max}, m, algorithm alg

Part 1: Strategy to store Message knowing $\boldsymbol{A}$;

Initialize the memory of alg to be $\mathbf{0}$.;

Submit $\emptyset$ to the Oracle and use the response as $\boldsymbol{v}_0$.;

Run alg with the optimization procedure knowing $\boldsymbol{A}$ and $\boldsymbol{v}_0$ until the first exploratory query $\boldsymbol{x}_{i_{1,1}}$.

for $p \in [p_{max}]$ **do**

  Let Memory_p be the current memory state of alg and $i_{p,1}$ the current iteration step. ;

  Run alg with the feasibility procedure until period p ends at iteration step $i_{p+1,1}$. If alg stopped before, **return** the strategy fails. When needed to sample a unit vector $\boldsymbol{v}_{p',l'}$, submit vectors $\boldsymbol{x}_{i_{p',1}}, \dots \boldsymbol{x}_{i_{p',r'}}$ to the Oracle where $i_{p',1}, \dots, i_{p',r'}$ are the exploratory queries defined at that stage. We use the corresponding response of the Oracle as $\boldsymbol{v}_{p',l'}$.;

  if $i_{p+1,1} - i_{p,1} \leq m$ **then**

    Set $\mathsf{Message} = \mathsf{Memory}_p$

end

for *Remaining queries to perform to Oracle* **do** Submit arbitrary query, e.g. $\emptyset$;

if Message *has not been defined yet* **then return** The strategy fails;

Submit $\boldsymbol{g}_{\boldsymbol{A},\boldsymbol{v}}$ to the Oracle as defined in Eq (7.7).;

Part 2: Strategy to make queries;

Set the memory state of alg to be Message and define $i_1 = 1$, $r = 1$.;

for $i \in [m]$ **do**

  Run alg with current memory to obtain a query $\boldsymbol{z}_i$.;

  Submit $\boldsymbol{z}_i$ to the Oracle from Game 7.2, to get response $(\boldsymbol{g}_i, s_i)$.;

  Compute $(v_i, \tilde{\boldsymbol{g}}_i)$ using $\boldsymbol{z}_i$, $\boldsymbol{g}_i$ and s_i as defined in Eq (7.8) and pass $(v_i, \tilde{\boldsymbol{g}}_i)$ as response to alg.;

  if $v_i \leq -\eta\gamma_1/2$ *and* $\|P_{Span(\boldsymbol{z}_{i_{r'}}, r' \leq r)^\perp}(\boldsymbol{z}_i)\|/\|\boldsymbol{z}_i\| \geq \frac{\gamma_2}{4}$ **then**

    Set $i_{r+1} = i$ and increment $r \leftarrow r + 1$.

end

Part 3: Strategy to return vectors;

if *index* i_k *has not been defined yet* **then**

  With the current memory of alg find a new query $\boldsymbol{z}_{m+1}$ and set $i_k = m + 1$.;

return $\left\{ \frac{\boldsymbol{z}_{i_1}}{\|\boldsymbol{z}_{i_1}\|}, \dots, \frac{\boldsymbol{z}_{i_k}}{\|\boldsymbol{z}_{i_k}\|} \right\}$ to the Oracle.

Algorithm 7.3: Strategy of the Player for the Orthogonal Vector Game with Hints

Next, if l is the index of the last constructed vector $\boldsymbol{v}_{p,l}$ before $i_{p,u}$ in the optimization procedure, one has $F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_{i_{p,u}}) \leq -\eta\gamma_1/2$. Therefore, $\|\boldsymbol{A}\boldsymbol{x}_{i_{p,u}}\|_\infty \leq F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_{i_{p,u}}) + \eta \leq \eta$. Further, $\eta \boldsymbol{v}_0^\top \boldsymbol{x}_{i_{p,u}} \leq F_{\boldsymbol{A},\boldsymbol{v},p,l}(\boldsymbol{x}_{i_{p,u}}) \leq -\eta\gamma_1/2$. This proves that $\|\boldsymbol{x}_{i_{p,u}}\| \geq \gamma_1/2$. Putting the previous two inequalities together yields

$$\frac{\|\boldsymbol{A}\boldsymbol{x}_{i_{p,u}}\|_\infty}{\|\boldsymbol{x}_{i_{p,u}}\|} \leq \frac{2\eta}{\gamma_1}.$$

As a result, this shows that the returned vectors are successful for Game 7.2 for the desired parameters $\alpha = 2\eta/\gamma_1$ and $\beta = \gamma_2/4$. Thus, the player wins on $\mathcal{E} \cap \mathcal{F}$, which has probability at least $q - (C + C')\sqrt{\log d}/d^2$ by the union bound. This ends the proof of the proposition. ■

7.3.6 Query lower bound for the Orthogonal Vector Game with Hints

Before proving a lower bound on the necessary number of queries for Game 7.2, we need to introduce two results. The first one is a known concentration result for vectors in the hypercube. It shows that for a uniform vector in the hypercube, being approximately orthogonal to k orthonormal vectors has exponentially small probability in k.

Lemma 7.3 ([Mar+22]). *Let $\boldsymbol{h} \sim \mathcal{U}(\{\pm 1\}^d)$. Then, for any $t \in (0, 1/2]$ and any matrix $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_k] \in \mathbb{R}^{d \times k}$ with orthonormal columns,*

$$\mathbb{P}(\|\boldsymbol{Z}^\top \boldsymbol{h}\|_\infty \leq t) \leq 2^{-c_H k}.$$

We will also need an anti-concentration bound for random vectors, which intuitively provides a lower bound for the previous concentration result. The following lemma shows that for a uniformly random unit vector, being orthogonal to k orthonormal vectors is still achievable with exponentially small probability in k.

Lemma 7.4. *Let $k < d$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ be k orthonormal vectors. Then,*

$$\mathbb{P}_{\boldsymbol{y} \sim \mathcal{U}(S^{d-1})}\left(|\boldsymbol{x}_i^\top \boldsymbol{y}| \leq \frac{1}{d^3}, \forall i \leq k\right) \geq \frac{1}{e^{d^{-4}} d^{3k}}.$$

Proof Let $\boldsymbol{y} \sim \mathcal{U}(S^{d-1})$ be a uniformly random unit vector. Then, for $i < k$ and any $y_1, \dots, y_{i-1}$ such that $|y_1|, \dots, |y_{i-1}| \leq \frac{1}{d^3}$, we have

$$\begin{aligned}
\mathbb{P}\left(|y_i| \leq \frac{1}{d^3} \mid y_1, \dots, y_{i-1}\right) &= \mathbb{P}_{\boldsymbol{u} \sim \mathcal{U}(S^{d-i})}\left(|u_1| \leq \frac{1}{d^3\sqrt{1 - (y_1^2 + \ldots + y_{i-1}^2)}}\right) \\
&\geq \frac{\int_0^{1/d^3} (1 - y^2)^{(d-i-1)/2} dy}{\int_0^1 (1 - y^2)^{(d-i-1)/2} dy} \\
&\geq \frac{(1 - d^{-6})^{d/2}}{d^3} \geq \frac{e^{-d^{-5}}}{d^3},
\end{aligned}$$

where in the last equation we used $d \geq 2$. Therefore, we can show by induction that $\mathbb{P}(|y_i| \leq 1/d^3, \forall i \leq k) \geq \frac{e^{-kd^{-5}}}{d^{3k}}$. Thus, by isometry this shows that

$$\mathbb{P}\left(|\boldsymbol{x}_i^\top \boldsymbol{y}| \leq \frac{1}{d^3}, \forall i \leq k\right) \geq \frac{1}{e^{d^{-4}} d^{3k}}.$$

This ends the proof of the lemma. ■

We are now ready to prove the query lower bound for Game 7.2 given in Proposition 7.4. Precisely, we show that for an appropriate choice of parameters, one needs $m = \tilde{\Omega}(d)$ queries. The proof is closely inspired from the arguments given in [Mar+22]. The main added difficulty arises from bounding the information leakage of the provided hints. As such, our goal is to show that these do not provide more information than the message itself.

Proof of Proposition 7.4 We first define some notations. Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_k]$ be the matrix storing the final outputs from the algorithm. Next, for the responses of the oracle $(\boldsymbol{g}_1, s_1), \dots, (\boldsymbol{g}_m, s_m)$, we first store all the scalar responses in a vector $\boldsymbol{c} = [s_1, \dots, s_m]$. We then focus on the responses $\boldsymbol{g}_1, \dots, \boldsymbol{g}_m$. Let $\tilde{\boldsymbol{G}}$ denote the matrix containing these responses of the oracle which are lines of $\boldsymbol{A}$. Let $\boldsymbol{G}$ be the matrix containing unique columns from $\tilde{\boldsymbol{G}}$, augmented with rows of $\boldsymbol{A}$ so that it has exactly m columns which are all different rows of $\boldsymbol{A}$. Last, let $\boldsymbol{A}'$ be the matrix $\boldsymbol{A}$ once the rows from $\boldsymbol{G}$ are removed. Next, let $\tilde{\boldsymbol{V}}$ be a matrix containing the responses of the oracle which are vectors $\boldsymbol{v}_l$, ordered by increasing index l. As before, let $\boldsymbol{V}$ be the matrix $\tilde{\boldsymbol{V}}$ where we only conserve unique columns and append it with additional vectors $\boldsymbol{v}_l$ so that $\boldsymbol{V}$ has exactly m columns. We denote by $\boldsymbol{w}_1, \dots, \boldsymbol{w}_m$ these vectors, and recall that they are vectors $\boldsymbol{v}_l$ ordered by increasing order of index l. Last, we define a vector $\boldsymbol{j}$ of indices such that $j(i)$ contains the information of which column of the matrices $\boldsymbol{G}$ or $\boldsymbol{V}$ corresponds $\boldsymbol{g}_i$. Precisely, if $\boldsymbol{g}_i$ is a line $\boldsymbol{a}$ from $\boldsymbol{A}$, we set $j(i) = j$ where j is the index of the column from $\boldsymbol{G}$ corresponding to $\boldsymbol{a}$. Otherwise, if j is the index of the column from $\boldsymbol{V}$ corresponding to $\boldsymbol{g}_i$, we set $j(i) = m + j$.

Next, we argue that $\boldsymbol{Y}$ is a deterministic function of $\mathsf{Message}$, the matrices $\boldsymbol{G}$, $\boldsymbol{V}$ and the vector of indices $\boldsymbol{j}$ and $\boldsymbol{c}$. First, $\boldsymbol{c}$ provides the scalar responses directly. For the d-dimensional component of the responses, first, note that from $\boldsymbol{G}$, $\boldsymbol{V}$ and $\boldsymbol{j}$ one can easily recover the vectors $\boldsymbol{g}_1, \dots, \boldsymbol{g}_m$. Next, using the algorithm for the second section of the Orthogonal Vector Game with Hints set with initial memory $\mathsf{Message}$ and the vectors $\boldsymbol{g}_1, \dots, \boldsymbol{g}_m$ as responses of the oracle, one can inductively compute the queries $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$. Last, $\boldsymbol{Y}$ is a deterministic function of $\boldsymbol{x}_i, \boldsymbol{g}_i, i \in [m]$ and $\mathsf{Message}$. This ends the claim that there is a function ϕ such that $\boldsymbol{Y} = \phi(\mathsf{Message}, \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c})$. By the data processing inequality,

$$I(\boldsymbol{A}'; \boldsymbol{Y} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) \leq I(\boldsymbol{A}'; \mathsf{Message} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) \leq H(\mathsf{Message} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) \leq M. \quad (7.9)$$

In the last inequality we used the fact that $\mathsf{Message}$ uses at most M bits. We have that

$$I(\boldsymbol{A}'; \boldsymbol{Y} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) = H(\boldsymbol{A}' \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) - H(\boldsymbol{A}' \mid \boldsymbol{Y}, \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}). \quad (7.10)$$

In the next steps we bound the two terms. We start with the second term of the right hand side of Eq (7.10) using similar arguments to the proof given in [Mar+22]. Let $\mathcal{E}$ be the event when the Player succeeds at Game 7.2. Consider the case when $\boldsymbol{Y}$ is a winning matrix. Then we have $\|\boldsymbol{A}\boldsymbol{y}_i\|_\infty \leq \alpha$ for all $i \leq k$. As a result, any line $\boldsymbol{a}$ of $\boldsymbol{A}'$ satisfies $\|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty \leq \alpha$. Further, we have that $\|P_{Span(\boldsymbol{y}_j, j<i)^\perp}(\boldsymbol{y}_i)\| \leq \beta$ for all $i \leq k$. By Lemma 7.6, there exist $\lceil k/5 \rceil$ orthonormal vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_{\lceil k/5 \rceil}]$ such that for any $\boldsymbol{x} \in \mathbb{R}^d$ one has $\|\boldsymbol{Z}^\top \boldsymbol{x}\|_\infty \leq \left(\frac{\sqrt{d}}{\beta}\right)^{5/4} \|\boldsymbol{Y}^\top \boldsymbol{x}\|_\infty$. In particular, all lines $\boldsymbol{a}$ of $\boldsymbol{A}'$ satisfy

$$\|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \left(\frac{\sqrt{d}}{\beta}\right)^{5/4} \alpha \leq \frac{1}{2},$$

where we used the hypothesis in the parameters α and β. By Lemma 7.3, one has

$$\left|\left\{\boldsymbol{a} \in \{\pm 1\}^d : \|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \le \frac{1}{2}\right\}\right| \le 2^d \mathbb{P}_{\boldsymbol{h}\sim\mathcal{U}(\{\pm 1\}^d)}\left(\|\boldsymbol{Z}^\top \boldsymbol{h}\|_\infty \le \frac{1}{2}\right) \le 2^{d - c_H \lceil k/5 \rceil}.$$

Therefore, we proved that if $\boldsymbol{Y}'$ is a winning vector, $H(\boldsymbol{A}' \mid \boldsymbol{Y} = \boldsymbol{Y}') \le (n-m)(d - c_H k/5)$. Otherwise, if $\boldsymbol{Y}'$ loses, we can directly use $H(\boldsymbol{A}' \mid \boldsymbol{Y} = \boldsymbol{Y}') \le (n-m)d$. Combining these equations gives

$$\begin{aligned} H(\boldsymbol{A}' \mid \boldsymbol{Y}, \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) &\le H(\boldsymbol{A}' \mid \boldsymbol{Y}) \\ &\le \mathbb{P}(\mathcal{E}^c)(n-m)d + \mathbb{P}(\mathcal{E})(n-m)(d - c_H k/5) \\ &\le (n-m)(d - \mathbb{P}(\mathcal{E}) c_H k/5). \end{aligned}$$

Next, we turn to the first term of the right-hand side of Eq (7.10).

$$\begin{aligned} H(\boldsymbol{A}' \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) = H(\boldsymbol{A} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}, \boldsymbol{c}) &= H(\boldsymbol{A} \mid \boldsymbol{V}) - I(\boldsymbol{A}; \boldsymbol{G}, \boldsymbol{j}, \boldsymbol{c} \mid \boldsymbol{V}) \\ &\ge H(\boldsymbol{A} \mid \boldsymbol{V}) - H(\boldsymbol{G}, \boldsymbol{j}, \boldsymbol{c}) \\ &\ge H(\boldsymbol{A} \mid \boldsymbol{V}) - md - m\log(2m) - m\log(d^2) \\ &= H(\boldsymbol{A}) - I(\boldsymbol{A}; \boldsymbol{V}) - md - 3m\log(2d) \\ &= (n-m)d - 3m\log(2d) - I(\boldsymbol{A}; \boldsymbol{V}). \end{aligned}$$

In the second inequality, we use the fact that $\boldsymbol{G}$ uses md bits and $\boldsymbol{j}$ can be stored with $m\log(2m)$ bits. By the chain rule,

$$I(\boldsymbol{A}; \boldsymbol{V}) = \sum_{i \le m} I(\boldsymbol{A}; \boldsymbol{w}_i \mid \boldsymbol{w}_1, \dots, \boldsymbol{w}_{i-1}).$$

Next, if $\boldsymbol{w}_i = \boldsymbol{v}_l$, recalling that the vectors $\boldsymbol{w}_{i'} = \boldsymbol{v}_{l'}$ are ordered by increasing index of l', we have

$$\begin{aligned} I(\boldsymbol{A}; \boldsymbol{w}_i \mid \boldsymbol{w}_1, \dots, \boldsymbol{w}_{i-1}) &= H(\boldsymbol{w}_i \mid \boldsymbol{w}_1, \dots, \boldsymbol{w}_{i-1}) - H(\boldsymbol{w}_i \mid \boldsymbol{A}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_i) \\ &\le H(\boldsymbol{w}_i) - H(\boldsymbol{w}_i \mid \boldsymbol{A}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_i, \boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,r_l}) \\ &= \log|\mathcal{D}_\delta| - H(\boldsymbol{w}_i \mid \boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,r_l}). \end{aligned}$$

In the last equality, we used the fact that if $\boldsymbol{b}_{l,1}, \dots, \boldsymbol{b}_{l,r_l}$ are the resulting vectors from the Gram-Schmidt decomposition of $\boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,r_l}$, $\boldsymbol{y}_l$ is generated uniformly in $S^{d-1} \cap \{\boldsymbol{y} : \forall r \le r_l, |\boldsymbol{b}_{l,r}^\top \boldsymbol{y}| \le d^{-3}\}$ independently from the past history, and $\boldsymbol{v}_l = \phi_\delta(\boldsymbol{y}_l)$. By Lemma 7.4, we know that

$$\mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}(S^{d-1})}\left(\forall r \le r_l, |\boldsymbol{b}_{l,r}^\top \boldsymbol{z}| \le d^{-3}\right) \ge \frac{1}{e^{d^{-4}} d^{3k}}.$$

As a result, for any $\boldsymbol{b}_j(\delta) \in \mathcal{D}_\delta$, one has

$$\mathbb{P}(\boldsymbol{w}_i = \boldsymbol{b}_j(\delta) \mid \boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,r_l}) \le \frac{\mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}(S^{d-1})}(\boldsymbol{z} \in V_j(\delta))}{\mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}(S^{d-1})}\left(\forall r \le r_l, |\boldsymbol{b}_{l,r}^\top \boldsymbol{z}| \le d^{-3}\right)} \le \frac{e^{d^{-4}} d^{3k}}{|\mathcal{D}_\delta|},$$

where we used the fact that each cell has the same area. In particular, this shows that

$$H(\boldsymbol{w}_i \mid \boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,r_l}) = \mathbb{E}_{\boldsymbol{b}\sim\boldsymbol{w}_i \mid \boldsymbol{x}_{l,1},\dots,\boldsymbol{x}_{l,r_l}}[-\log p_{\boldsymbol{w}_i \mid \boldsymbol{x}_{l,1},\dots,\boldsymbol{x}_{l,r_l}}(\boldsymbol{b})] \ge \log\left(\frac{|\mathcal{D}_\delta|}{e^{d^{-4}} d^{3k}}\right).$$

Hence,

$$I(\boldsymbol{A}; \boldsymbol{w}_i \mid \boldsymbol{w}_1, \dots, \boldsymbol{w}_{i-1}) \leq 3k \log d + d^{-4} \log e.$$

Putting everything together gives

$$\begin{aligned}
&I(\boldsymbol{A}'; \boldsymbol{Y} \mid \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{j}) \\
&\geq (n-m)d - 3m\log(2d) - 3km\log d - 2md^{-4} - (n-m)(d - \mathbb{P}(\mathcal{E})c_H k/5) \\
&\geq \frac{c_H}{10}k(n-m) - 3km\log d - 1 - 3d\log(2d),
\end{aligned}$$

where in the last equation we used $d \geq 2$. Together with Eq (7.9), this implies

$$m \geq \frac{c_H kn/10 - M - 1 - 3d\log(2d)}{k(3\log d + c_H/10)}.$$

As a result, since $k \geq 20\frac{M+3d\log(2d)+1}{c_H n}$ and $n \geq d/4$, we obtain

$$m \geq \frac{c_H n}{60\log d + 2c_H} \geq \frac{c_H}{8(30\log d + c_H)}d.$$

This ends the proof of the proposition. ■

7.4 Memory-constrained feasibility problem

In this section, we prove the lower bound from Theorem 7.2 for the feasibility problem.

7.4.1 Defining the feasibility procedure

Similarly to Section 7.3, we pose $n = \lceil d/4 \rceil$. Also, for any matrix $\boldsymbol{A} \in \{\pm 1\}^{n \times d}$, we use the same functions $\boldsymbol{g}_{\boldsymbol{A}}$ and $\tilde{\boldsymbol{g}}_{\boldsymbol{A}}$. We use similar techniques as those we introduced for the optimization problem. However, since in this case, the separation oracle only returns a separating hyperplane, without any value considerations of an underlying function, Procedure 7.1 can be drastically simplified, which leads to improved lower bounds.

Let $\eta_0 = 1/(24d^2)$, $\eta_1 = \frac{1}{2\sqrt{d}}$, $\delta = 1/d^3$, and $k \leq d/3 - n$ be a parameter. Last, let $p_{max} = \lfloor (c_{d,1}d - 1)/(k-1) \rfloor$, where $c_{d,1}$ is the same quantity as in Eq (7.3). The feasibility procedure is defined in Procedure 7.4. The oracle first randomly samples $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n \times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$. This matrix and vector are then fixed in the rest of the procedure. Whenever the player queries a point $\boldsymbol{x}$ such that $\|\boldsymbol{A}\boldsymbol{x}\|_\infty > \eta_0$ (resp. $\boldsymbol{v}_0^\top \boldsymbol{x} > -\eta_1$), the oracle returns $\tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x})$ (resp. $\boldsymbol{v}_0$). All other queries are called *informative* queries. With this definition, it remains to define the separation oracle on informative queries. The oracle proceeds by periods in which the behavior is different. In each period p, the oracle constructs vectors $\boldsymbol{v}_{p,1}, \dots, \boldsymbol{v}_{p,k-1}$ inductively and keeps in memory some queries $i_{p,1}, \dots, i_{p,k}$ that will be called *exploratory*. The first informative query t will be the first exploratory query and starts period 1.

Given a new query $\boldsymbol{x}_t$,

1. If $\|\boldsymbol{A}\boldsymbol{x}\|_\infty > \eta_0$, the oracle returns $\tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t)$.

2. If $\boldsymbol{v}_0^\top \boldsymbol{x}_t > -\eta_1$, the oracle returns $\boldsymbol{v}_0$.

3. If $\boldsymbol{x}_t$ was queried in the past sequence, the oracle returns the same vector that was returned previously.

4. Otherwise, let p be the index of the current period and let $\boldsymbol{v}_{p,1}, \dots, \boldsymbol{v}_{p,l}$ be the vectors from the current period constructed so far, together with their corresponding exploratory queries $i_{p,1} \dots, i_{p,l} < t$. Potentially, if $p = 1$ one may not have defined any such vectors at the beginning of time t. In this case, let $l = 0$.

   (a) If $\max_{1 \le l' \le l} \boldsymbol{v}_{p,l'}^\top \boldsymbol{x}_t > -\eta_1$ (with the convention $\max_\emptyset = -\infty$), the oracle returns $\boldsymbol{v}_{p,l'}$ where $l' = \arg\max_{l \le r} \boldsymbol{v}_{p,l}^\top \boldsymbol{x}_t$. Ties are broken alphabetically.

   (b) Otherwise, if $l < k - 1$, we first define $i_{p,l+1} = t$. Then, let $\boldsymbol{b}_{p,1}, \dots, \boldsymbol{b}_{p,l+1}$ be the result from the Gram-Schmidt decomposition of $\boldsymbol{x}_{i_{p,1}}, \dots, \boldsymbol{x}_{i_{p,l+1}}$ and let $\boldsymbol{y}_{p,l+1}$ be a sample of the distribution obtained by the uniform distribution $\boldsymbol{y}_{p,l+1} \sim \mathcal{U}(S^{d-1} \cap \left\{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p,r}^\top \boldsymbol{z}| \le \frac{1}{d^3}, \forall r \le l+1\right\})$. We then pose $\boldsymbol{v}_{p,l+1} = \phi_\delta(\boldsymbol{y}_{p,l+1})$. Having defined this new vector, the oracle returns $\boldsymbol{v}_{p,l+1}$. We then increment l.

   (c) Otherwise, if $l = k$, we define $i_{p,k} = i_{p+1,1} = t$. If $p + 1 \le p_{max}$, this starts the next period $p+1$. As above, let $\boldsymbol{b}_{p+1,1}$ be the result of the Gram-Schmidt decomposition of $\boldsymbol{x}_{i_{p+1,1}}$ and sample $\boldsymbol{y}_{p+1,1}$ according to a uniform $\boldsymbol{y}_{p+1,1} \sim \mathcal{U}(S^{d-1} \cap \left\{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p+1,1}^\top \boldsymbol{z}| \le \frac{1}{d^3}\right\})$. We then pose $\boldsymbol{v}_{p+1,1} = \phi_\delta(\boldsymbol{y}_{p+1,1})$ and the oracle returns $\boldsymbol{v}_{p+1,1}$. We can then increment p and reset $l = 1$.

The above construction ends when the period p_{max} is finished. At this point, the oracle has defined the vectors $\boldsymbol{v}_{p,l}$ for all $p \le p_{max}$ and $l \le k$. We then define the successful set as

$$Q_{\boldsymbol{A},\boldsymbol{v}} = \left\{\boldsymbol{x} \in B_d(\boldsymbol{0}, 1) : \|\boldsymbol{A}\boldsymbol{x}\|_\infty \le \eta_0, \boldsymbol{v}_0^\top \boldsymbol{x} \le -\eta_1, \max_{p \le p_{max}, l \le k-1} \boldsymbol{v}_{p,l}^\top \boldsymbol{x} \le -\eta_1\right\}.$$

From now on, the procedure uses any separation oracle for $Q_{\boldsymbol{A},\boldsymbol{v}}$ as responses to the algorithm, while making sure to be consistent with previous oracle reponses if a query is exactly duplicated. We next define what we mean by solving the above feasibility procedure.

Definition 7.3. *Let alg be an algorithm for the feasibility problem. When running alg with the responses of the feasibility procedure, we denote by $\boldsymbol{v}$ the set of constructed vectors and $\boldsymbol{x}^\star(alg)$ the final answer returned by alg. We say that an algorithm alg is successful for the feasibility procedure with probability $q \in [0, 1]$, if taking $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n \times d})$, with probability at least q over the randomness of $\boldsymbol{A}$ and of the procedure, $\boldsymbol{x}^\star(alg) \in Q_{\boldsymbol{A},\boldsymbol{v}}$.*

In the rest of this section, we first relate this feasibility procedure to the standard feasibility problem, then prove query lower bounds to solve the feasibility procedure.

Input: d, k, p_{max}, algorithm alg

Sample $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n\times d})$ and $\boldsymbol{v}_0 \sim \mathcal{U}(\mathcal{D}_\delta)$.;
Initialize the memory of alg to $\mathbf{0}$ and let $p = 1$, $l = 0$.;
for $t \geq 1$ **do**
    Run alg with current memory to obtain a query $\boldsymbol{x}_t$;
    if $\|\boldsymbol{A}\boldsymbol{x}_t\| > \eta_0$ **then return** $\tilde{\boldsymbol{g}}_{\boldsymbol{A}}(\boldsymbol{x}_t)$ as response to alg ;
    else if $\boldsymbol{v}_0^\top \boldsymbol{x}_t > -\eta_1$ **then return** $\boldsymbol{v}_0$ as response to alg ;
    else if *Query* $\boldsymbol{x}_t$ *was made in the past* **then return** the same vector that was returned for $\boldsymbol{x}_t$;
    else
        if $\max_{1\leq l'\leq l} \boldsymbol{v}_{p,l'}^\top \boldsymbol{x}_t > -\eta_1$ **then**
            return $\boldsymbol{v}_{p,l'}$ where $l' = \arg\max_{l\leq r} \boldsymbol{v}_{p,l}^\top \boldsymbol{x}_t$.
        else if $l < k-1$ **then**
            Let $i_{p,l+1} = t$ and compute Gram-Schmidt decomposition $\boldsymbol{b}_{p,1}, \dots, \boldsymbol{b}_{p,l+1}$ of $\boldsymbol{x}_{i_{p,1}}, \dots, \boldsymbol{x}_{i_{p,l+1}}$.;
            Sample $\boldsymbol{y}_{p,l+1}$ uniformly on $\mathcal{S}^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p,l'}^\top \boldsymbol{z}| \leq d^{-3}, \forall l' \leq l+1\}$ and define $\boldsymbol{v}_{p,l+1} = \phi_\delta(\boldsymbol{y}_{p,l+1})$.;
            return $\boldsymbol{v}_{p,l+1}$ as response to alg and increment $l \leftarrow l+1$.
        else if $p+1 \leq p_{max}$ **then**
            Set $i_{p,k} =, i_{p+1,1} = t$ and compute the Gram-Schmidt decomposition $\boldsymbol{b}_{p+1,1}$ of $\boldsymbol{x}_{i_{p+1,1}}$.;
            Sample $\boldsymbol{y}_{p+1,1}$ uniformly on $\mathcal{S}^{d-1} \cap \{\boldsymbol{z} \in \mathbb{R}^d : |\boldsymbol{b}_{p+1,1}^\top \boldsymbol{z}| \leq d^{-3}\}$ and define $\boldsymbol{v}_{p+1,1} = \phi_\delta(\boldsymbol{y}_{p+1,1})$.
            return $\boldsymbol{v}_{p+1,1}$ as response to alg, increment $p \leftarrow p+1$ and reset $l = 1$.
        else Set $i_{p_{max},k} = t$ and break the **for** loop;
end
for $t' \geq t$ **do** Use any separation oracle for $Q_{\boldsymbol{A},\boldsymbol{v}}$ consistent with previous responses ;

Procedure 7.4: The feasibility procedure for algorithm alg

7.4.2 Reduction from the feasibility problem to the feasibility procedure

In the next proposition, we check that the above procedure indeed corresponds to a valid feasibility problem.

Proposition 7.6. *On an event of probability at least* $1 - C\sqrt{\log d}/d$*, the procedure described above is a valid feasibility problem. More precisely, the following hold.*

- *There exists* $\bar{\boldsymbol{x}} \in B_d(\mathbf{0}, 1)$ *such that* $\|\boldsymbol{A}\bar{\boldsymbol{x}}\|_\infty = 0$, $\boldsymbol{v}_0^\top \bar{\boldsymbol{x}} \leq -4\eta_1$, *and*

$$\max_{p\leq p_{max}, l\leq k-1} \boldsymbol{v}_{p,l}^\top \bar{\boldsymbol{x}} \leq -4\eta_1.$$

- *Let* $\epsilon = \min\{\eta_0/\sqrt{d}, \eta_1\}/2$. *Then,* $B_d\left(\bar{\boldsymbol{x}} - \epsilon\frac{\bar{\boldsymbol{x}}}{\|\bar{\boldsymbol{x}}\|}, \epsilon\right) \subseteq B_d(\mathbf{0}, 1) \cap B_d(\bar{\boldsymbol{x}}, 2\epsilon) \subseteq Q_{\boldsymbol{A},\boldsymbol{v}}$.

- *Throughout the run of the feasibility problem, the separation oracle always returned a valid cut, i.e., for any iteration t, if $\boldsymbol{x}_t$ denotes the query and $\boldsymbol{g}_t$ is the returned vector from the oracle, one has*

$$\forall \boldsymbol{x} \in Q_{\boldsymbol{A},\boldsymbol{v}}, \quad \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x} \rangle > 0.$$

Further, responses are consistent: if $\boldsymbol{x}_t = \boldsymbol{x}_{t'}$, the responses of the procedure at times t and t' coincide.

Proof We use a similar proof to that of Proposition 7.5. For convenience, we rename $\boldsymbol{v}_{p,l} = \boldsymbol{v}_{(p-1)(k-1)+l}$. Also, let $l_{max} = p_{max}(k-1) \leq c_{d,1}d - 1$. Next, let $C_d = \sqrt{40 l_{max} \log d}$. We define the vector

$$\bar{\boldsymbol{x}} = -\frac{1}{C_d} \sum_{l=0}^{l_{max}} P_{Span(\boldsymbol{a}_i, i \leq n)^\perp}(\boldsymbol{v}_l).$$

Since $l_{max} \leq p_{max}(k-1) \leq c_{d,1}d - 1$, the same arguments as in the proof of Proposition 7.5 show that on an event $\mathcal{E}$ of probability at least $1 - C\sqrt{\log d}/d$, we have $\|\bar{\boldsymbol{x}}\| \leq 1$ and

$$\max_{0 \leq l \leq l_{max}} \boldsymbol{v}_l^\top \bar{\boldsymbol{x}} \leq -\frac{1}{40\sqrt{(l_{max}+1)\log d}} \leq -\frac{2}{\sqrt{d}} = -4\eta_1,$$

where in the second inequality we used $l_{max} \leq c_{d,1}d - 1$. By construction, one has $\|\boldsymbol{A}\bar{\boldsymbol{x}}\|_\infty = 0$. This ends the proof of the first claim of the proposition. We then turn to the second claim, which is immediate from the fact that $\boldsymbol{x} \mapsto \|\boldsymbol{A}\boldsymbol{x}\|_\infty$ is $\sqrt{d}$-Lipschitz and both $\boldsymbol{x} \mapsto \boldsymbol{v}_0^\top \boldsymbol{x}$ and $\boldsymbol{x} \mapsto \max_{p \leq p_{max}, l \leq k} \boldsymbol{v}_{p,l}^\top \boldsymbol{x}$ are 1-Lipschitz. Therefore, $B_d(\bar{\boldsymbol{x}} - \epsilon\bar{\boldsymbol{x}}/\|\bar{\boldsymbol{x}}\|, \epsilon) \subseteq B_d(\boldsymbol{0}, 1) \cap B_d(\bar{\boldsymbol{x}}, 2\epsilon) \subset Q_{\boldsymbol{A},\boldsymbol{v}}$. It remains to check that the third claim is satisfied. It suffices to check that this is the case during the construction phase of the feasibility procedure. By construction of $Q_{\boldsymbol{A},\boldsymbol{v}} \subset \{\boldsymbol{x} : \|\boldsymbol{A}\boldsymbol{x}\|_\infty \leq \eta_0\}$.

Hence, it suffices to check that for informative queries $\boldsymbol{x}_t$, the returned vectors $\boldsymbol{g}_t$ are valid separation hyperplanes. By construction, these can only be either $\boldsymbol{v}_0$ or $\boldsymbol{v}_{p,l}$ for $p \leq p_{max}$, $l \leq k-1$. We denote by $\boldsymbol{w}$ this vector. Let t' be the first time $\boldsymbol{x}_t$ was queried. There are two cases. Either $\boldsymbol{w}$ was not constructed at time t', in which case, by construction this means that we are in scenario (2) or (4a). Both cases imply $\boldsymbol{w}^\top \boldsymbol{x}_t > -\eta_1$. Hence, $\boldsymbol{w}$ which is returned by the procedure is a valid separation hyperplane. Now suppose that $\boldsymbol{w} = \boldsymbol{v}_{p,l}$ was constructed at time t'—scenarios (4b) or (4c). By construction, one has $|\boldsymbol{b}_{p,r}^\top \boldsymbol{y}_{p,l}| \leq d^{-3}$ for all $r \leq l$. Decomposing $\boldsymbol{x}_t = \boldsymbol{x}_{i_{p,l}} = \alpha \boldsymbol{b}_{p,1} + \ldots + \alpha_l \boldsymbol{b}_{p,l}$, we obtain

$$|\boldsymbol{x}_t^\top \boldsymbol{y}_{p,l}| \leq \frac{\|\boldsymbol{\alpha}\|_1}{d^3} \leq \frac{1}{d^2\sqrt{d}}.$$

As a result, $\boldsymbol{y}_{p,l}^\top \boldsymbol{x}_t \geq -1/(d^2\sqrt{d})$. Because $\boldsymbol{v}_{p,l} = \phi_\delta(\boldsymbol{y}_{p,l})$, we have $\|\boldsymbol{v}_{p,l} - \boldsymbol{y}_{p,l}\| \leq \delta$. Hence, for any $d \geq 2$,

$$\boldsymbol{w}^\top \boldsymbol{x}_t \geq -1/(d^2\sqrt{d}) - \delta > -\eta_1.$$

Hence, $\boldsymbol{w}$ was a valid separation hyperplane. The last claim that the responses of the procedure are consistent over time is a direct consequence from its construction. This ends the proof of the proposition. ■

As a simple consequence of this result, solving the feasibility problem is harder than solving the feasibility procedure with high probability.

Proposition 7.7. *Let alg be an algorithm that solves the feasibility problem with accuracy* $\epsilon = 1/(48d^2\sqrt{d})$. *Then, it solves the feasibility procedure with probability at least* $1 - C\sqrt{\log d}/d$.

Proof Let $\mathcal{E}$ be the event of probability at least $1 - C\sqrt{\log d}/d$ defined in Proposition 7.6. We show that on $\mathcal{E}$, *alg* solves the feasibility procedure. On $\mathcal{E}$, the feasibility procedure emulates is a valid feasibility oracle. Further, on $\mathcal{E}$, the successful set contains a closed ball of radius ϵ. As a result, on $\mathcal{E}$, *alg* finds a solution to the feasibility problem emulated by the procedure. ■

Next, we show that it is necessary to finish the p_{max} periods to solve the feasibility procedure.

Proposition 7.8. *Fix an algorithm alg. Then, if* $\mathcal{E}$ *denotes the event when alg succeeds and* $\mathcal{B}$ *denotes the event when the procedure ends period* p_{max} *with alg, then* $\mathcal{E} \subseteq \mathcal{B}$.

Proof Consider the case when the period p_{max} was not ended. Let $\boldsymbol{x}^\star$ denote the last query performed by *alg*. We consider the scenario in which $\boldsymbol{x}^\star$ fell. Let t be the first time when *alg* submitted query $\boldsymbol{x}^\star$. For any of the scenarios (1), (2), or (4a), by construction of $Q_{\boldsymbol{A},\boldsymbol{v}}$, we already have $\boldsymbol{x}_t \notin Q_{\boldsymbol{A},\boldsymbol{v}}$. It remains to check scenarios (4b) and (4c) for which the procedure constructs a new vector $\boldsymbol{v}_{p,l}$, where p is the index of the period of t and $i_{p,1}, \ldots, i_{p,l} = t$ are the previous exploratory queries in period p. We decompose $\boldsymbol{x}_t = \boldsymbol{x}_{i_{p,l}} = \alpha_1 \boldsymbol{b}_{p,1} + \alpha_l \boldsymbol{b}_{p,l}$. By construction,

$$|\boldsymbol{x}_t^\top \boldsymbol{y}_{p,l}| = |\boldsymbol{x}_{i_{p,l}}^\top \boldsymbol{y}_{p,l}| \leq \frac{\|\boldsymbol{\alpha}\|_1}{d^3} \leq \frac{1}{d^2\sqrt{d}}.$$

As a result, $\boldsymbol{x}_t^\top \boldsymbol{v}_{p,l} \geq -|\boldsymbol{x}_t^\top \boldsymbol{y}_{p,l}| - \delta \geq -d^{-2.5} - d^{-3} > -\eta_1$, for any $d \geq 2$. Thus, $\boldsymbol{x}_t = \boldsymbol{x}^\star \notin Q_{\boldsymbol{A},\boldsymbol{v}}$. This shows that in order to succeed at the feasibility procedure, an algorithm needs to end all p_{max} periods. ■

7.4.3 Reduction to the Orthogonal Vector Game with Hints.

The remaining piece of our argument is to show that solving the feasibility procedure is harder than solving the Orthogonal Vector Game with Hints, Game 7.2.

Proposition 7.9. *Let* $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{n \times d})$. *If there exists an* M*-bit algorithm that solves the feasibility problem described above using* mp_{max} *queries with probability at least* q *over the randomness of the algorithm, choice of* $\boldsymbol{A}$ *and the randomness of the separation oracle, then there is an algorithm for Game 7.2 for parameters* $(d, k, m, M, \alpha = \frac{\eta_0}{\eta_1}, \beta = \frac{\eta_1}{2})$, *for which the Player wins with probability at least* q *over the randomness of the player's strategy and* $\boldsymbol{A}$.

Proof Let *alg* be an M-bit algorithm solving the feasibility problem with mp_{max} queries with probability at least q. In Algorithm 7.5, we describe the strategy of the player in Game 7.2.

In the first part of the strategy, the player observes $\boldsymbol{A}$. Then they proceed to simulate the feasibility problem with *alg* using parameters $\boldsymbol{A}$. When needed to sample a vector $\boldsymbol{v}_{p,l}$ (resp. $\boldsymbol{v}_0$), the player submits the corresponding queries $\boldsymbol{x}_{i_{p,1}}, \ldots, \boldsymbol{x}_{i_{p,l}}$ (resp. $\emptyset$) useful to define

Input: d, k, p_{max}, m, algorithm alg

Part 1: Strategy to store Message knowing $\boldsymbol{A}$;

Initialize the memory of alg to be $\mathbf{0}$.;
Submit $\emptyset$ to the Oracle and use the response as $\boldsymbol{v}_0$.;
Run alg with the optimization procedure knowing $\boldsymbol{A}$ and $\boldsymbol{v}_0$ until the first exploratory query $\boldsymbol{x}_{i_{1,1}}$.
for $p \in [p_{max}]$ **do**
    Let Memory_p be the current memory state of alg and $i_{p,1}$ the current iteration step. ;
    Run alg with the feasibility procedure until period p ends at iteration step $i_{p+1,1}$. If alg stopped before, **return** the strategy fails. When needed to sample a unit vector $\boldsymbol{v}_{p',l'}$, submit vectors $\boldsymbol{x}_{i_{p',1}}, \dots \boldsymbol{x}_{i_{p',l'}}$ to the Oracle. We use the corresponding response of the Oracle as $\boldsymbol{v}_{p',l'}$.;
    if $i_{p+1,1} - i_{p,1} \leq m$ **then**
        Set $\mathsf{Message} = \mathsf{Memory}_p$
end
for *Remaining queries to perform to Oracle* **do** Submit arbitrary query, e.g. $\emptyset$;
if Message *has not been defined yet* **then return** The strategy fails;
Submit $\tilde{\boldsymbol{g}}_{\boldsymbol{A},\boldsymbol{v}}$ to the Oracle as defined in Eq (7.11).;

Part 2: Strategy to make queries;

Set the memory state of alg to be Message.;
for $i \in [m]$ **do**
    Run alg with current memory to obtain a query $\boldsymbol{z}_i$.;
    Submit $\boldsymbol{z}_i$ to the Oracle from Game 7.2, to get response $(\boldsymbol{g}_i, s_i)$.;
    Compute $\tilde{\boldsymbol{g}}_i$ using $\boldsymbol{z}_i$, $\boldsymbol{g}_i$ and s_i as defined in Eq (7.12) and pass $\tilde{\boldsymbol{g}}_i$ as response to alg.;
end

Part 3: Strategy to return vectors;

for $l \in [k]$ **do** Set i_l to be the index i of the first query $\boldsymbol{z}_i$ for which $s_i = l$, if it exists ;
if *index* i_k *has not been defined yet* **then**
    With the current memory of alg find a new query $\boldsymbol{z}_{m+1}$ and set $i_k = m + 1$.;
return $\left\{ \frac{\boldsymbol{z}_{i_1}}{\|\boldsymbol{z}_{i_1}\|}, \dots, \frac{\boldsymbol{z}_{i_k}}{\|\boldsymbol{z}_{i_k}\|} \right\}$ to the Oracle.

Algorithm 7.5: Strategy of the Player for the Orthogonal Vector Game with Hints

$\boldsymbol{v}_{p,l}$. The player then takes the response given by the Oracle as that vector $\boldsymbol{v}_{p,l}$ (resp. $\boldsymbol{v}_0$), which simulates exactly a run of the feasibility procedure. Further, since $1 + p_{max}(k-1) \leq d$, the player does not run out of queries. Importantly, during the run, the player keeps track of the length $i_{p,k} - i_{p,1}$ of period p. The first time we encounter a period p with length at most m, we set $\mathsf{Message} = \mathsf{Memory}_p$, the memory state of alg at the beginning of period p. If there is no such period, the strategy fails. Also, if alg stopped before ending period p_{max}, the strategy fails. Next, the algorithm submits the following function $\tilde{\boldsymbol{g}}_{\boldsymbol{A},\boldsymbol{v}}$ to the Oracle. Since the responses of the feasibility procedure are consistent over time, we adopt the following notation. For a previously queried vector $\boldsymbol{x}$ of alg, we denote $\boldsymbol{g}(\boldsymbol{x})$ the vector which was

returned to alg during the first part (lines 3-9 of Algorithm 7.5).

$$\tilde{\boldsymbol{g}}_{\boldsymbol{A},\boldsymbol{v}} : \boldsymbol{x} \mapsto \begin{cases} (\boldsymbol{0}, 1) & \text{if } \boldsymbol{x} \text{ was never queried in the first part,} \\ (\boldsymbol{a}_i, 1) & \text{ow. and if } \boldsymbol{g}(\boldsymbol{x}) \in \{\pm \boldsymbol{a}_i\}, i \leq n, \\ (\boldsymbol{v}_0, 2) & \text{ow. and if } \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{v}_0, \\ (\boldsymbol{v}_{p',l'}, 2 + l' \mathbb{1}_{p'=p} + k \mathbb{1}_{p'=p+1,l'=1}) & \text{ow. if } \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{v}_{p',l'}, p' \leq p_{max}, l \leq k-1. \end{cases} \tag{7.11}$$

Intuitively, the first component of $\tilde{\boldsymbol{g}}$ gives the returned vector in the first period, at the exception that we always return $\boldsymbol{a}_i$ instead of $\{\pm \boldsymbol{a}_i\}$. The second term has values in $[2+k \leq d^2]$. Hence, the submitted function is valid.

Next, in the second part of the algorithm, the player proceeds to simulate a run the feasibility procedure with alg on period p. To do so, we first set the memory state of alg to Message. Each new query $\boldsymbol{z}_i$ is submitted to the Oracle of Game 7.2 to get a response $(\boldsymbol{g}_i, s_i)$. Then, we compute $\tilde{\boldsymbol{g}}_i$ as follows

$$\tilde{\boldsymbol{g}}_i = \begin{cases} \boldsymbol{g}_i & \text{if } s_i \geq 2, \\ sign(\boldsymbol{g}_i^\top \boldsymbol{z}_i) \boldsymbol{g}_i & \text{if } s_i = 1. \end{cases} \tag{7.12}$$

One can easily check that $\tilde{\boldsymbol{g}}_i$ corresponds exactly to the response that was passed to alg in the first part of the strategy. The player then passes $\tilde{\boldsymbol{g}}_i$ to alg so that it can update its state. We repeat this process for m steps. Further, the player can also keep track of the exploratory queries: the index i_l of the first response satisfying $s_i = 2 + l$ for $l \leq k-1$ (resp. $s_i = 2+k$)is the exploratory query which led to the construction of $\boldsymbol{v}_{p,l}$ (resp. $\boldsymbol{v}_{p+1,1}$) in the first part. Last, we check if the last index i_k was defined. If not, we pose $i_k = m+1$ and let $\boldsymbol{z}_{m+1}$ be the next query of alg with the current memory. The player then returns the vectors $\frac{\boldsymbol{z}_{i_1}}{\|\boldsymbol{z}_{i_1}\|}, \dots, \frac{\boldsymbol{z}_{i_k}}{\|\boldsymbol{z}_{i_k}\|}$. This ends the description of the player's strategy.

By Proposition 7.8, on an event $\mathcal{E}$ of probability at least q, the algorithm alg succeeds and ends period p_{max}. As a result, similarly as in the proof of Proposition 7.3, since alg makes at most mp_{max} queries, and there are p_{max} periods, there must be a period of length at most m. Hence the strategy never fails at this phase of the player's strategy on the event $\mathcal{E}$. Further, we already checked that in the second phase, the vectors $\tilde{\boldsymbol{g}}_i$ passed to alg coincide exactly with the responses passed to alg in the first part. Thus, this shows that during the second part, the player simulates exactly the run of the feasibility problem on period p. More precisely, the queries coincide with the queries in the feasibility problem at times $i_{p,1}, \dots, \min\{i_{p,k}, i_{p,1}+m-1\}$. Because the first part succeeded on $\mathcal{E}$, we have $i_{p,k} \leq i_{p,0}+m$. Therefore, if i_k has not yet been defined, this means that we had $i_{p,k} = i_{p,1} + m$. Hence, the next query with the current memory $\boldsymbol{z}_{m+1}$ is exactly the query $\boldsymbol{x}_{i_{p,k}}$ for the feasibility problem. This shows that the vectors $\boldsymbol{z}_{i_1}, \dots, \boldsymbol{z}_{i_k}$ coincide exactly with the vectors $\boldsymbol{x}_{i_{p,1}}, \dots, \boldsymbol{x}_{i_{p,k}}$ when running alg on the feasibility problem in the first part.

We now show that the returned vectors are successful for Game 7.2. By construction, $\boldsymbol{x}_{i_{p,1}}, \dots, \boldsymbol{x}_{i_{p,k}}$ are all informative. In particular, $\|\boldsymbol{A}\boldsymbol{x}_{i_{p,l}}\|_\infty \leq \eta_0$ for all $1 \leq l \leq k$. Further, these queries did not fall in scenario (2), hence $\boldsymbol{v}_0^\top \boldsymbol{x}_{i_{p,l}} < -\eta_1$, which implies $\|\boldsymbol{x}_{i_{p,l}}\| > \eta_1$ for all $l \leq k$. As a result,

$$\frac{\|\boldsymbol{A}\boldsymbol{x}_{i_{p,l}}\|_\infty}{\|\boldsymbol{x}_{i_{p,l}}\|} \leq \frac{\eta_0}{\eta_1}.$$

Next fix $l \leq k-1$. By construction of $\boldsymbol{y}_{p,l}$,

$$\|P_{Span(\boldsymbol{x}_{i_{p,l'}}, l' \leq l)}(\boldsymbol{y}_{p,l})\|^2 = \sum_{l' \leq l} |\boldsymbol{b}_{p,l'}^\top \boldsymbol{y}_{p,l}|^2 \leq \frac{k}{d^6} \leq \frac{1}{d^5}.$$

Hence,

$$\|\boldsymbol{v}_{p,l} - P_{Span(\boldsymbol{x}_{i_{p,l'}}, l' \leq l)^\perp}(\boldsymbol{y}_{p,l})\| \leq \|P_{Span(\boldsymbol{x}_{i_{p,l'}}, l' \leq l)}(\boldsymbol{y}_{p,l})\| + \delta \leq \frac{1}{d^5} + \delta.$$

As a result, since $\boldsymbol{x}_{p,l+1}^\top \boldsymbol{v}_{p,l} < -\eta_1$, we have

$$\|P_{Span(\boldsymbol{x}_{i_{p,l'}}, l' \leq l)^\perp}(\boldsymbol{x}_{p,l+1})\| \geq |\boldsymbol{x}_{p,l+1}^\top P_{Span(\boldsymbol{x}_{i_{p,l'}}, l' \leq l)^\perp}(\boldsymbol{y}_{p,l})\| > \eta_1 - \frac{1}{d^5} - \delta \geq \frac{\eta_1}{2}.$$

This shows that the returned vectors $\frac{\boldsymbol{x}_{i_{p,1}}}{\|\boldsymbol{x}_{i_{p,1}}\|}, \ldots, \frac{\boldsymbol{x}_{i_{p,k}}}{\|\boldsymbol{x}_{i_{p,k}}\|}$ are successful for Game 7.2 with parameters $\alpha = \frac{\eta_0}{\eta_1}$ and $\beta = \frac{\eta_1}{2}$. This ends the proof that strategy succeeds on $\mathcal{E}$ for these parameters, which ends the proof of the proposition. ■

We are now ready to prove the main result.

Proof of Theorem 7.2 Suppose that there is an algorithm alg for solving the feasibility problem to optimality $\epsilon = 1/(48d^2\sqrt{d})$ with memory M and at most Q queries. Let $k = \lceil 20\frac{M+3d\log(2d)+1}{c_H n} \rceil$. By Proposition 7.7, it solves the feasibility procedure with parameter k with probability at least $1 - C\sqrt{\log d}/d$. By Proposition 7.9 there is an algorithm for Game 7.2 that wins with probability 1/3 with $m = \lceil Q/p_{max} \rceil$ and paraeters $\alpha = \eta_0/\eta_1$ and $\beta = \eta_1/2$. We check that

$$\alpha \left(\frac{\sqrt{d}}{\beta}\right)^{5/4} \leq 12d^2\eta_0 = \frac{1}{2}.$$

Hence, by Proposition 7.4, we have

$$m \geq \frac{c_H}{8(30\log d + c_H)} d.$$

This shows that

$$Q \geq \Omega\left(p_{max}\frac{d}{\log d}\right) = \Omega\left(\frac{d^2}{k\log^3 d}\right) = \Omega\left(\frac{d^3}{(M+\log d)\log^3 d}\right).$$

This implies that for a memory $M = d^{2-\delta}$ with $0 \leq \delta \leq 1$ the number of queries is $Q = \tilde{\Omega}(d^{1+\delta})$. ■

7.5 Appendix

7.5.1 Concentration bounds

The following result gives concentration bounds for the norm of the projection of a random unit vector onto linear subspaces.

Proposition 7.10. *Let P be a projection in $\mathbb{R}^d$ of rank r and let $\boldsymbol{x} \in \mathbb{R}^d$ be a random vector sampled uniformly on the unit sphere $\boldsymbol{x} \sim \mathcal{U}(S^{d-1})$. Then, for every $t > 0$,*

$$\max\left\{\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \geq t\right), \mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \leq -t\right)\right\} \leq e^{-dt^2}.$$

Further, if $r = 1$ and $d \geq 2$,

$$\mathbb{P}\left(\|P(\boldsymbol{x})\| \geq \sqrt{\frac{t}{d-1}}\right) \leq 2\sqrt{t}e^{-t/2}.$$

Proof First, by isometry, we can assume that P is the projection onto the coordinate vectors $\boldsymbol{e}_1, \dots \boldsymbol{e}_r$. Then, let $\boldsymbol{y} \sim \mathcal{N}(0,1)$ be a normal vector. Note that $\boldsymbol{x} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} \sim \mathcal{U}(S^{d-1})$. Further,

$$\|\boldsymbol{x}\|^2 \geq \frac{r}{d} + t \iff \left(1 - \frac{r}{d} - t\right)\sum_{i=1}^{r} y_i^2 \geq \left(\frac{r}{d} + t\right)\sum_{i=r+1}^{d} y_i^2.$$

Note that $Z_1 = \sum_{i=1}^{r} y_i^2$ and $Z_2 = \sum_{i=r+1}^{d} y_i^2$ are two independent random chi squared variables of parameters r and $d-r$ respectively. Recalling that the moment generating function of $Z \sim \chi^2(k)$ is $\mathbb{E}[e^{sZ}] = (1-2s)^{-k/2}$ for $s < 1/2$. Therefore, for any

$$-\frac{1}{2(r/d+t)} < s < \frac{1}{2(1-r/d-t)}, \tag{7.13}$$

one has

$$\begin{aligned}\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \geq t\right) &\leq \mathbb{E}\left[\exp\left(s\left(1 - \frac{r}{d} - t\right)Z_1 - s\left(\frac{r}{d} + t\right)Z_2\right)\right] \\ &= \frac{\left[1 - 2s\left(1 - \frac{r}{d} - t\right)\right]^{-r/2}}{\left[1 - 2s\left(\frac{r}{d} + t\right)\right]^{-(d-r)/2}}.\end{aligned}$$

Let $s = \frac{1}{2}\left(\frac{1-r/d}{1-r/d-t} - \frac{r/d}{r/d+t}\right)$, which satisfies Eq (7.13). The previous equation readily yields

$$\mathbb{P}\left(\left|\|P(\boldsymbol{x})\|^2 - \frac{r}{d}\right| \geq t\right) \leq \exp\left(-\frac{d}{2}d_{KL}\left(\frac{r}{d}; \frac{r}{d} + t\right)\right) \leq e^{-dt^2}.$$

In the last inequality we used Pinsker's inequality $d_{KL}(r/d; r/d + t) \geq 2\delta(\mathcal{B}(r/d), \mathcal{B}(d/r + t))^2 = 2t^2$, where $\mathcal{B}(q)$ is the Bernoulli distribution of parameter q. Replacing P with $Id - P$ and r with $d - r$ gives the other inequality

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \leq -t\right) \leq e^{-dt^2}.$$

This gives first claim. For the second claim, supposing that $r = 1 < d$, from the above equation, we have

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \geq \frac{t}{d}\right) \leq \exp\left(-\frac{d}{2}d_{KL}\left(\frac{1}{d}; \frac{t}{d}\right)\right) = \sqrt{t}\left(\frac{1 - \frac{t}{d}}{1 - \frac{1}{d}}\right)^{(d-1)/2} \leq \sqrt{2t}e^{-t(d-1)/(2d)}.$$

Thus,

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \geq \frac{t}{d-1}\right) \leq \sqrt{\frac{2(d-1)}{d}}\sqrt{t}e^{-t/2},$$

which ends the proof of the proposition. ■

Next, we need the following lemma which gives a concentration inequality for discretized samples in $\mathcal{D}_d$ and approximately perpendicular to $k \leq d/3-1$ vectors.

Lemma 7.5. *Let $0 \leq k \leq d/3-1$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k \in B_d(\mathbf{0}, 1)$ be k orthonormal vectors in the unit ball, and $\boldsymbol{x} \in B_d(\mathbf{0}, 1)$. Denote by μ the distribution on the unit sphere corresponding to the uniform distribution $\boldsymbol{y} \sim \mathcal{U}(S^{d-1} \cap \{\boldsymbol{w} \in \mathbb{R}^d : |\boldsymbol{x}_i^\top \boldsymbol{w}| \leq d^{-3}, \forall i \leq k\})$. Let $\boldsymbol{y} \sim \mu$. Then, for $t \geq 2$,*

$$\mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{y}| \geq \sqrt{\frac{t}{d}} + \frac{1}{d^2}\right) \leq 2\sqrt{t}e^{-t/3}.$$

Further, let $\delta \leq 1$ and $\boldsymbol{z} = \phi_\delta(\boldsymbol{y})$. Then for $t \geq 4$,

$$\mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{z}| \geq \sqrt{\frac{t}{d}} + \frac{1}{d^2} + \delta\right) \leq 2\sqrt{t}e^{-t/3}.$$

Proof We use the same notations as above and denote by $\mathcal{E} = \{|\boldsymbol{x}_i^\top \boldsymbol{y}| \leq d^{-3}, \forall i \leq k\}$ the event considered and $\boldsymbol{y} \sim \mu$. We decompose $\boldsymbol{y} = \alpha_1 \boldsymbol{x}_1 + \ldots + \alpha_k \boldsymbol{x}_k + \boldsymbol{y}'$, where $\boldsymbol{y}' \in Span(\boldsymbol{x}_i, i \leq k)^\perp := E$. Note that $\frac{\boldsymbol{y}'}{\|\boldsymbol{y}'\|}$ is a uniformly random unit vector in E. As a result, using Proposition 7.10, we obtain for any $t \geq 2$,

$$\begin{aligned}\mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{y}'| \geq \sqrt{\frac{t}{d-k-1}}\right) &= \mathbb{P}\left(|P_E(\boldsymbol{x})^\top \boldsymbol{y}'| \geq \sqrt{\frac{t}{d-k-1}}\right) \\ &\leq 2\sqrt{t}e^{-t/2}.\end{aligned}$$

Also, because by definition of μ, we have $|\alpha_i| \leq d^{-3}$ for all $i \leq k$, we obtain $|\boldsymbol{x}^\top \boldsymbol{y}| \leq \frac{k}{d^3} + |\boldsymbol{x}^\top \boldsymbol{y}'| \leq \frac{1}{d^2} + |\boldsymbol{x}^\top \boldsymbol{y}'|$. As a result, using the fact that $d-k-1 \geq 2d/3$, the previous equation shows that

$$\mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{y}| \geq \sqrt{\frac{3t}{2d}} + \frac{1}{d^2}\right) \leq \mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{y}'| \geq \sqrt{\frac{t}{d-k-1}}\right) \leq 2\sqrt{t}e^{-t/2}.$$

Next, we use the fact that $\|\boldsymbol{z} - \boldsymbol{y}\| = \|\phi_\delta(\boldsymbol{y}) - \boldsymbol{y}\| \leq \delta$ to obtain

$$\mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{z}| \geq \sqrt{\frac{t}{d}} + \frac{1}{d^2} + \delta\right) \leq \mathbb{P}\left(|\boldsymbol{x}^\top \boldsymbol{y}| \geq \sqrt{\frac{t}{d}} + \frac{1}{d^2}\right) \leq 2\sqrt{t}e^{-t/3}.$$

This ends the proof of the lemma. ■

7.5.2 Robustly-independent vectors

The following lemma serves the same purpose as [Mar+22, Lemma 34]. Namely, from successful vectors of the Game 7.2, it allows to recover an orthonormal basis that is still approximately in the nullspace of $\boldsymbol{A}$. The following version gives a stronger version that improves the dependence in d of our chosen parameters.

Lemma 7.6. *Let $\delta \in (0, 1]$ and suppose that we have $r \leq d$ unit norm vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r \in \mathbb{R}^d$. Suppose that for any $i \leq k$,*

$$\|P_{Span(\boldsymbol{y}_j, j<i)^\perp}(\boldsymbol{y}_i)\| \geq \delta.$$

Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r]$ and $s \geq 2$. There exists $\lceil r/s \rceil$ orthonormal vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{\lceil r/s \rceil}]$ such that for any $\boldsymbol{a} \in \mathbb{R}^d$,

$$\|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \left(\frac{\sqrt{d}}{\delta}\right)^{s/(s-1)} \|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty.$$

Proof Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r)$ be the orthonormal basis given by the Gram-Schmidt decomposition of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r$. By definition of the Gram-Schmidt decomposition, we can write $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{C}$ where $\boldsymbol{C}$ is an upper-triangular matrix. Further, its diagonal is exactly given via $diag(\|P_{Span(\boldsymbol{y}_{l'}, l'<l)^\perp}(\boldsymbol{y}_l)\|, l \leq r)$. Hence,

$$\det(\boldsymbol{Y}) = \det(\boldsymbol{C}) = \prod_{l \leq r} \|P_{Span(\boldsymbol{y}_{l'}, l'<l)^\perp}(\boldsymbol{y}_l)\| \geq \delta^r.$$

We then introduce the singular value decomposition $\boldsymbol{Y} = \boldsymbol{U} diag(\sigma_1, \ldots, \sigma_r)\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times r}$ have orthonormal columns, and $\sigma_1 \geq \ldots \geq \sigma_r$. Next, for any vector $\boldsymbol{z} \in \mathbb{R}^d$, since the columns of $\boldsymbol{Y}$ have unit norm,

$$\|\boldsymbol{Y}\boldsymbol{z}\|_2 \leq \sum_{l \leq r} |z_l| \|\boldsymbol{y}_l\|_2 \leq \|\boldsymbol{z}\|_1 \leq \sqrt{d}\|\boldsymbol{z}\|_2.$$

In the last inequality we used Cauchy-Schwartz. Therefore, all singular values of $\boldsymbol{Y}$ are upper bounded by $\sigma_1 \leq \sqrt{d}$. Thus, with $r' = \lceil r/s \rceil$

$$\delta^r \leq \det(\boldsymbol{Y}) = \prod_{l=1}^{r} \sigma_l \leq d^{(r'-1)/2} \sigma_{r'}^{r-r'+1} \leq d^{r/2s} \sigma_{r'}^{(s-1)r/s},$$

so that $\sigma_{r'} \geq \delta^{s/(s-1)}/d^{1/(2s)}$. We are ready to define the new vectors. We pose for all $i \leq r'$, $\boldsymbol{z}_i = \boldsymbol{u}_i$ the i-th column of $\boldsymbol{U}$. These correspond to the r' largest singular values of $\boldsymbol{Y}$ and are orthonormal by construction. Then, for any $i \leq r'$, we also have $\boldsymbol{z}_i = \boldsymbol{u}_i = \frac{1}{\sigma_i}\boldsymbol{Y}\boldsymbol{v}_i$ where $\boldsymbol{v}_i$ is the i-th column of $\boldsymbol{V}$. Hence, for any $\boldsymbol{a} \in \mathbb{R}^d$,

$$|\boldsymbol{z}_i^\top \boldsymbol{a}| = \frac{1}{\sigma_i}|\boldsymbol{v}_i^\top \boldsymbol{Y}^\top \boldsymbol{a}| \leq \frac{\|\boldsymbol{v}_i\|_1}{\sigma_i}\|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty \leq \frac{d^{1/2+1/(2s)}}{\delta^{s/(s-1)}}\|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty.$$

This ends the proof of the lemma. ■

Chapter 8

Memory-Constrained Algorithms for Convex Optimization

8.1 Introduction

Optimization algorithms are ubiquitous in machine learning, from solving simple regressions to training neural networks. Their essential roles have motivated numerous studies on their efficiencies, which are usually analyzed through the lens of oracle-complexity: given an oracle (such as function value, or subgradient oracle), how many calls to the oracle are needed for an algorithm to output an approximate optimal solution? [NYD83]. However, ever-growing problem sizes have shown an inadequacy in considering only the oracle-complexity, and have motivated the study of the trade-off between oracle-complexity and other resources such as memory [WS19; Mar+22] and communication [LLZ20; Red+16; SSZ14; Smi+17; Mot+13; ZDW12; Wan+18; WWS17].

In this work, we study the oracle-complexity/memory trade-off for first-order non-smooth convex optimization, and the closely related feasibility problem, with a focus on developing memory efficient (deterministic) algorithms. Since [WS19] formally posed as open problem the question of characterizing this trade-off, there have been exciting results showing what is impossible: for convex optimization in $\mathbb{R}^d$, [Mar+22] shows that any randomized algorithm with $d^{1.25-\delta}$ bits of memory needs at least $\tilde{\Omega}(d^{1+4\delta/3})$ queries, and we showed in Chapter 7 that this can be improved for deterministic algorithms to $d^{1-\delta}$ bits of memory or $\tilde{\Omega}(d^{1+\delta/3})$ queries; in addition we showed that for the feasibility problem with a separation oracle, any algorithm which uses $d^{2-\delta}$ bits of memory needs at least $\tilde{\Omega}(d^{1+\delta})$ queries.

Despite these recent results on the lower bounds, all known first-order convex optimization algorithms that output an ϵ-suboptimal point fall into two categories: those that have quadratic memory in the dimension d but can potentially achieve the optimal $\mathcal{O}(d \ln \frac{1}{\epsilon})$ query complexity, as represented by the center-of-mass method, and those that have $\mathcal{O}(\frac{1}{\epsilon^2})$ query complexity but only need the optimal $\mathcal{O}(d \ln \frac{1}{\epsilon})$ bits of memory, as represented by the classical gradient descent [WS19]. In addition, the above-mentioned memory bounds apply only between queries, and in particular the center-of-mass method [WS19] is allowed to use infinite memory during computations.

We propose a family of memory-constrained algorithms for the stronger feasibility prob-

lem in which one aims to find a point within a set Q containing a ball of radius ϵ, with access to a separation oracle. In particular, this can be used for convex optimization since the subgradient information provides a separation vector. Our algorithms use $\mathcal{O}(\frac{d^2}{p}\ln\frac{1}{\epsilon})$ bits of memory (including during computations) and $\mathcal{O}((C\frac{d}{p}\ln\frac{1}{\epsilon})^p)$ queries for some universal constant $C \geq 1$, and a parameter $p \in [d]$ that can be chosen by the user. Intuitively, in the context of convex optimization, the algorithms are based on the idea that for any function $f(\boldsymbol{x}, \boldsymbol{y})$ convex in the pair $(\boldsymbol{x}, \boldsymbol{y})$, the partial minimum $\min_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$ as a function of $\boldsymbol{x}$ is still convex and, using a variant of Vaidya's method proposed in [LSW15], our algorithm can approximate subgradients for that function $\min_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$, thereby turning an optimization problem with variables $(\boldsymbol{x}, \boldsymbol{y})$ to one with just $\boldsymbol{x}$. This idea, applied recursively with the variables divided into p blocks, gives our family of algorithms and the above-mentioned memory and query complexity. The main algorithmic contribution is in how we design the recursive dimension reduction procedure: a technical step of the design and analysis is to ensure that the necessary precision for recursive computations can be achieved using low memory. Last, our algorithms account for memory usage throughout computations, as opposed to simply between calls to the gradient oracle, which was the traditional approach in the literature.

When $p = 1$, our algorithm is a memory-constrained version of Vaidya's method [Vai96; LSW15], and improves over the center-of-mass [WS19] method by a factor of $\ln\frac{1}{\epsilon}$ in terms of memory while having optimal oracle-complexity. The improvements provided by our algorithms are more significant in regimes when ϵ is very small in the dimension d: increasing the parameter p can further reduce the memory usage of Vaidya's method ($p = 1$) by a factor $\ln\frac{1}{\epsilon}/\ln d$, while still improving over the oracle-complexity of gradient descent. In particular, in a regime $\ln\frac{1}{\epsilon} = \text{poly}(\ln d)$, these memory improvements are only in terms of $\ln d$ factors. However, in sub-polynomial regimes with potentially $\ln\frac{1}{\epsilon} = d^c$ for some constant $c > 0$, these provide polynomial improvements to the memory of standard cutting-plane methods.

As a summary, this chapter makes the following contributions.

- Our class of algorithms provides a trade-off between memory-usage and the oracle-complexity whenever $\ln\frac{1}{\epsilon} \gg \ln d$. Further, taking $p = 1$ improves the memory-usage from center-of-mass [WS19] by a factor $\ln\frac{1}{\epsilon}$, while preserving the optimal oracle-complexity.

- For $\ln\frac{1}{\epsilon} \geq \Omega(d\ln d)$, our algorithm with $p = d$ is the first known algorithm that outperforms gradient descent in terms of the oracle-complexity, but still maintains the optimal $\mathcal{O}(d\ln\frac{1}{\epsilon})$ memory usage.

- We show how to obtain a $\ln\frac{1}{\epsilon}$ dependence in the known lower-bound trade-offs from [Mar+22] and Chapter 7, confirming that the oracle-complexity/memory trade-off is necessary for any regime $\epsilon \lesssim \frac{1}{\sqrt{d}}$.

8.2 Setup and Preliminaries

In this section, we precise the formal setup for our results. We follow the framework introduced in [WS19], to define the memory constraint on algorithms with access to an oracle

$\mathcal{O} : \mathcal{S} \to \mathcal{R}$ which takes as input a query $q \in \mathcal{S}$ and outputs a response $\mathcal{O}(q) \in \mathcal{R}$. Here, the algorithm is constrained to update an internal M-bit memory between queries to the oracle.

Definition 8.1 (M-bit memory-constrained algorithm [WS19; Mar+22])**.** *Let $\mathcal{O} : \mathcal{S} \to \mathcal{R}$ be an oracle. An M-bit memory-constrained algorithm is specified by a query function $\psi_{query} : \{0,1\}^M \to \mathcal{S}$ and an update function $\psi_{update} : \{0,1\}^M \times \mathcal{S} \times \mathcal{R} \to \{0,1\}^M$. The algorithm starts with the memory state $\mathsf{Memory}_0 = 0^M$ and iteratively makes queries to the oracle. At iteration t, it makes the query $q_t = \psi_{query}(\mathsf{Memory}_{t-1})$ to the oracle, receives the response $r_t = \mathcal{O}(q_t)$ then updates its memory $\mathsf{Memory}_t = \psi_{update}(\mathsf{Memory}_{t-1}, q_t, r_t)$.*

The algorithm can stop at any iteration and the last query is its final output. Importantly, this model does not enforce constraints on the memory usage during the computation of ψ_{update} and ψ_{query}. This is ensured in the stronger notion of a memory-constrained algorithm with computations. These are precisely algorithms that have constrained memory including for computations, with the only specificity that they need a decoder function ϕ to make queries to the oracle from their bit memory, and a discretization function ψ to write a discretized response into the algorithm's memory.

Definition 8.2 (M-bit memory-constrained algorithm with computations)**.** *Let $\mathcal{O} : \mathcal{S} \to \mathcal{R}$ be an oracle. We suppose that we are given a decoding function $\phi : \{0,1\}^\star \to \mathcal{S}$ and a discretization function $\psi : \mathcal{R} \times \mathbb{N} \to \{0,1\}^\star$ such that $\psi(r,n) \in \{0,1\}^n$ for all $r \in \mathcal{R}$. An M-bit memory-constrained algorithm with computations is only allowed to use an M-bit memory in $\{0,1\}^M$ even during computations. The algorithm has three special memory placements Q, N, R. Say the contents of Q and N are q and n respectively. To make a query, R must contain at least n bits. The algorithm submits q to the encoder which then submits the query $\phi(q)$ to the oracle. If $r = \mathcal{O}(\phi(q))$ is the oracle response, the discretization function then writes $\psi(r,n)$ in the placement R.*

Feasibility problem. In this problem, the goal is to find a point $\boldsymbol{x} \in Q$, where $Q \subset \mathcal{C}_d := [-1,1]^d$ is a convex set. We choose the cube $[-1,1]^d$ as prior bound for convenience in our later algorithms, but the choice of norm for this prior ball can be arbitrary and does not affect our results. The algorithm has access to a *separation oracle* $O_S : \mathcal{C}_d \to \{\mathsf{Success}\} \cup \mathbb{R}^d$, such that for a query $\boldsymbol{x} \in \mathbb{R}^d$ either returns $\mathsf{Success}$ if $\boldsymbol{x} \in Q$, or a separating hyperplane $\boldsymbol{g} \in \mathbb{R}^d$, i.e., such that $\boldsymbol{g}^\top \boldsymbol{x} < \boldsymbol{g}^\top \boldsymbol{x}'$ for any $\boldsymbol{x}' \in Q$. We suppose that the separating hyperplanes are normalized, $\|\boldsymbol{g}\|_2 = 1$. An algorithm solves the feasibility problem with accuracy ϵ if the algorithm is successful for any feasibility problem such that Q contains an ϵ-ball $B_d(\boldsymbol{x}^\star, \epsilon)$ for $\boldsymbol{x}^\star \in \mathcal{C}_d$.

As an important remark, this formulation asks that the separation oracle is consistent over time: when queried at the exact same point $\boldsymbol{x}$, the oracle always returns the same separation vector. In this context, we can use the natural decoding function ϕ which takes as input d sequences of bits and outputs the vector with coordinates given by the sequences interpreted in base 2. Similarly, the natural discretization function ψ takes as input the separation hyperplane $\boldsymbol{g}$ and outputs a discretized version up to the desired accuracy. From now, we can omit these implementation details and consider that the algorithm can query the oracle for discretized queries $\boldsymbol{x}$, up to specified rounding errors.

Remark 8.1. *An algorithm for the feasibility problem with accuracy $\epsilon/(2\sqrt{d})$ can be used for first-order convex optimization. Suppose one aims to minimize a 1-Lipschitz convex function f over the unit ball, and output an ϵ-suboptimal solution, i.e., find a point $\boldsymbol{x}$ such that $f(\boldsymbol{x}) \leq \min_{\boldsymbol{y} \in B_d(0,1)} f(\boldsymbol{y}) + \epsilon$. A separation oracle for $Q = \{\boldsymbol{x} : f(\boldsymbol{x}) \leq \min_{\boldsymbol{y} \in B_d(0,1)} f(\boldsymbol{y}) + \epsilon\}$ is given at a query $\boldsymbol{x}$ by the subgradient information from the first-order oracle: $-\frac{\partial f(\boldsymbol{x})}{\|\partial f(\boldsymbol{x})\|}$. Its computation can also be carried out memory-efficiently up to rounding errors since if $\|\partial f(\boldsymbol{x})\| \leq \epsilon/(2\sqrt{d})$, the algorithm can return $\boldsymbol{x}$ and already has the guarantee that $\boldsymbol{x}$ is an ϵ-suboptimal solution ($\mathcal{C}_d$ has diameter $2\sqrt{d}$). Notice that because f is 1-Lipschitz, Q contains a ball of radius $\epsilon/(2\sqrt{d})$ (the factor $1/(2\sqrt{d})$ is due to potential boundary issues). Hence, it suffices to run the algorithm for the feasibility problem while keeping in memory the queried point with best function value.*

8.2.1 Known trade-offs between oracle-complexity and memory

Known lower-bound trade-offs. All known lower bounds apply to the more general class of memory-constrained algorithms without computational constraints given in Definition 8.1. [NYD83] first showed that $\mathcal{O}(d \ln \frac{1}{\epsilon})$ queries are needed for solving convex optimization to ensure that one finds an ϵ-suboptimal solution. Further, $\mathcal{O}(d \ln \frac{1}{\epsilon})$ bits of memory are needed even just to output a solution in the unit ball with ϵ accuracy [WS19]. These historical lower bounds apply in particular to the feasibility problem and are represented in the pictures of Fig. 8.1 as the dashed pink region.

More recently, [Mar+22] showed that achieving both optimal oracle-complexity and optimal memory is impossible for convex optimization. They show that a possibly randomized algorithm with $d^{1.25-\delta}$ bits of memory makes at least $\tilde{\Omega}(d^{1+4\delta/3})$ queries. We extended thi result for deterministic algorithms in Chapter 7 showing that a deterministic algorithm with $d^{1-\delta}$ bits of memory makes $\tilde{\Omega}(d^{1+\delta/3})$ queries. For the feasibility problem, we gave an improved trade-off: any deterministic algorithm with $d^{2-\delta}$ bits of memory makes $\tilde{\Omega}(d^{1+\delta})$ queries. These trade-offs are represented in the left picture of Fig. 8.1 as the pink, red, and purple solid region, respectively. Using a clever and more careful analysis,[CP23] showed that similar lower bounds can be carried out for deterministic algorithms as well.

Known upper-bound trade-offs. Prior to this work, to the best of our knowledge only two algorithms were known in the oracle-complexity/memory landscape. First, cutting-plane algorithms achieve the optimal oracle-complexity $\mathcal{O}(d \ln \frac{1}{\epsilon})$ but use quadratic memory. The memory-constrained (MC) center-of-mass method analyzed in [WS19] uses in particular $\mathcal{O}(d^2 \ln^2 \frac{1}{\epsilon})$ memory. Instead, if one uses Vaidya's method which only needs to store $\mathcal{O}(d)$ cuts instead of $\mathcal{O}(d \ln \frac{1}{\epsilon})$, we show that one can achieve $\mathcal{O}(d^2 \ln \frac{1}{\epsilon})$ memory. These algorithms only use the separation oracle and hence apply to both convex optimization and the feasibility problem. On the other hand, the memory-constrained gradient descent for convex optimization [WS19] uses the optimal $\mathcal{O}(d \ln \frac{1}{\epsilon})$ memory but makes $\mathcal{O}(\frac{1}{\epsilon^2})$ iterations. While the analysis in [WS19] is only carried for convex optimization, we can give a modified proof showing that gradient descent can also be used for the feasibility problem.

8.2.2 Other related works

Vaidya's method [Vai96; RM95; Ans97; Ans98] and the variant [LSW15] that we use in our algorithms, belong to the family of cutting-plane methods. Perhaps the simplest example of an algorithm in this family is the center-of-mass method, which achieves the optimal $\mathcal{O}(d \ln \frac{1}{\epsilon})$ oracle-complexity but is computationally intractable, and the only known random walk-based implementation [BV04] has computational complexity $\mathcal{O}(d^7 \ln \frac{1}{\epsilon})$. Another example is the ellipsoid method, which has suboptimal $\mathcal{O}(d^2 \ln \frac{1}{\epsilon})$ query complexity, but has an improved computational complexity $\mathcal{O}(d^4 \ln \frac{1}{\epsilon})$. [Bub15] pointed out that Vaidya's method achieves the best of both worlds by sharing the $\mathcal{O}(d \ln \frac{1}{\epsilon})$ optimal query complexity of the center-of-mass, and achieving a computational complexity of $\mathcal{O}(d^{1+\omega} \ln \frac{1}{\epsilon})$[1]. In a major breakthrough, this computational complexity was improved to $\mathcal{O}(d^3 \ln^3 \frac{1}{\epsilon})$ in [LSW15], then to $\mathcal{O}(d^3 \ln \frac{1}{\epsilon})$ in [Jia+20]. We refer to [Bub15; LSW15; Jia+20] for more detailed comparisons of these algorithms.

Another popular convex optimization algorithm that requires quadratic memory is the Broyden– Fletcher– Goldfarb– Shanno (BFGS) algorithm [Sha70; Bro70; Fle70; Gol70], which stores an approximated inverse Hessian matrix as gradient preconditioner. Several works aimed to reduce the memory usage of BFGS; in particular, the limited memory BFGS (L-BFGS) stores a few vectors instead of the entire approximated inverse Hessian matrix [Noc80; LN89]. However, it is still an open question whether even the original BFGS converges for non-smooth convex objectives [LO13].

Lying at the other extreme of the oracle-complexity/memory trade-off is gradient descent, which achieves the optimal memory usage but requires significantly more queries than center-of-mass or Vaidya's method in the regime $\epsilon \lesssim \frac{1}{\sqrt{d}}$. There is a rich literature of works aiming to speed up gradient descent, such as the optimized gradient method [DT14; DF14], Nesterov's Acceleration [Nes83], the triple momentum method [SFL17], geometric descent [BLS15], quadratic averaging [DFR18], the information-theoretic exact method [TD23], or Big-Step-Little-Step method [Kel+22]. Interested readers can find a comprehensive survey on acceleration methods in [dST21]. However, these acceleration methods usually require additional smoothness or strong convexity assumptions (or both) on the objective function, due to the known $\Omega(\frac{1}{\epsilon^2})$ query lower bound in the large-scale regime $\epsilon \gtrsim \frac{1}{\sqrt{d}}$ for any first order method where the query points lie in the span of the subgradients of previous query points [Nes03].

Besides accelerating gradient descent, researchers have investigated more efficient ways to leverage subgradients obtained in previous iterations. Of interest are bundle methods [BN05; Lan15; LNN95], that have found a wide range of applications [Teo+10; LSV07]. In their simplest form, they minimize the sum of the maximum of linear lower bounds constructed using past oracle queries, and a regularization term penalizing the distance from the current iteration variable. Although the theoretical convergence rate of the bundle method is the same as that of gradient descent, in practice, bundle methods can benefit from previous information and substantially outperform gradient descent [BN05].

Our works are focused on high-accuracy regimes, when the accuracy ϵ is sub-polynomial. We note that for their lower-bound result on randomized algorithms, [CP23] also required

[1] $\omega < 2.373$ is the exponent of matrix multiplication

sub-polynomial accuracies, which raises the question whether this is a general phenomenon for the study of memory-constrained algorithms in convex optimization. This also relates our work to the study of low-dimensional problems—or even constant dimension—which has been investigated in the literature [Vav93; BM20].

Last, the increasing size of optimization problems has also motivated the development of communication-efficient optimization algorithms in distributed settings such as [LLZ20; Red+16; SSZ14; Smi+17; Mot+13; ZDW12; Wan+18; WWS17]. Moreover, recent works have explored the trade-off between sample complexity and memory/communication complexity for learning problems under the streaming model, with notable contributions including [Bra+16; DKS19; DS18; Raz17; SSV19; MM17].

8.2.3 Outline of the chapter

In Section 8.3 we state our main results, in particular, we give the oracle-complexity and memory guarantees of our algorithms. In Section 8.4, we describe our algorithms without taking into account computational concerns and prove our memory and oracle-complexity guarantees. We our method without computational concerns, which already provides the main ideas. We then prove our complete results including computational concerns in Section 8.5. While previous lower-bounds results for oracle-complexity/memory trade-offs did not include any dependency in the accuracy ϵ, we give a general method in Section 8.6 to add a dependency $\ln \frac{1}{\epsilon}$ for both the oracle-complexity and memory usage. We conclude in Section 8.7. The proof of the query-complexity of gradient descent for feasibility problems is included in Section 8.8 for completeness.

8.3 Main Results

We first check that the memory-constrained gradient descent method solves feasibility problems. This was known for convex optimization [WS19] and the same algorithm with a modified proof gives the following result. For completeness, the proof is given in Section 8.8.

Proposition 8.1. *The memory-constrained gradient descent algorithm solves the feasibility problem with accuracy $\epsilon \leq \frac{1}{\sqrt{d}}$ using $\mathcal{O}(d \ln \frac{1}{\epsilon})$ bits of memory and $\mathcal{O}(\frac{1}{\epsilon^2})$ separation oracle calls.*

Our main contribution is a class of algorithms based on Vaidya's cutting-plane method that provide a query-complexity / memory trade-off for optimization in $\mathbb{R}^d$. More precisely, we show the following, where $\omega < 2.373$ is the exponent of matrix multiplication, such that multiplying two $n \times n$ matrices runs in $\mathcal{O}(n^\omega)$ time.

Theorem 8.1. *For any $1 \leq p \leq d$, there is a deterministic first-order algorithm that solves the feasibility problem in dimension d for accuracy $\epsilon \leq \frac{1}{\sqrt{d}}$, using $\mathcal{O}(\frac{d^2}{p} \ln \frac{1}{\epsilon})$ bits of memory (including during computations), with $\mathcal{O}((C\frac{d}{p} \ln \frac{1}{\epsilon})^p)$ calls to the separation oracle, and computational complexity $\mathcal{O}((C(\frac{d}{p})^{1+\omega} \ln \frac{1}{\epsilon})^p)$, where $C \geq 1$ is a universal constant.*

For simplicity, in Section 8.4, we describe algorithms that achieve this trade-off without computation concerns (Definition 8.1), which already provide the main elements of our

method. The proof of oracle-complexity and memory usage is given in Section 8.4.3. In Section 8.5, we consider computational constraints and give corresponding algorithms using the cutting-plane method of [LSW15].

To better understand the implications of Theorem 8.1, it is useful to compare the provided class of algorithms to the two algorithms known in the oracle-complexity/memory trade-off landscape: the memory-constrained center-of-mass method and the memory-constrained gradient descent [WS19].

For $p = 1$, our resulting procedure, which is essentially a memory-constrained Vaidya's algorithm, has optimal oracle-complexity $\mathcal{O}(d \ln \frac{1}{\epsilon})$ and uses $\mathcal{O}(d^2 \ln \frac{1}{\epsilon})$ bits of memory. This improves by a $\ln \frac{1}{\epsilon}$ factor the memory usage of the center-of-mass-based algorithm provided in [WS19], which used $\mathcal{O}(d^2 \ln^2 \frac{1}{\epsilon})$ memory and had the same optimal oracle-complexity.

Next, we recall that the memory-constrained gradient descent method used the optimal number $\mathcal{O}(d \ln \frac{1}{\epsilon})$ bits of memory (including for computations), and a sub-optimal $\mathcal{O}(\frac{1}{\epsilon^2})$ oracle-complexity. While the memory of our algorithms decreases with p, their oracle-complexity is exponential in p. This significantly restricts the values of p for which the oracle-complexity is improved over that of gradient descent. The range of application of Theorem 8.1 is given in the next result, where $\vee$ and $\wedge$ represent maximum and minimum respectively.

Corollary 8.1. *The algorithms given in Theorem 8.1 effectively provide a trade-off for* $p \leq \mathcal{O}(\frac{\ln \frac{1}{\epsilon}}{\ln d} \wedge d)$. *Precisely, this provides a trade-off between*

- *using* $\mathcal{O}(d^2 \ln \frac{1}{\epsilon})$ *memory with optimal* $\mathcal{O}(d \ln \frac{1}{\epsilon})$ *oracle-complexity, and*
- *using* $\mathcal{O}(d^2 \ln d \vee d \ln \frac{1}{\epsilon})$ *memory with* $\mathcal{O}(\frac{1}{\epsilon^2} \wedge (C \ln \frac{1}{\epsilon})^d)$ *oracle-complexity.*

Importantly, for $\epsilon \leq \frac{1}{d^{\Omega(d)}}$, taking $p = d$ yields an algorithm that uses the optimal memory $\mathcal{O}(d \ln \frac{1}{\epsilon})$ and has an improved query complexity over gradient descent. In this regime of small (virtually constant) dimension, for the same memory usage, gradient descent has a query complexity that is polynomial in ϵ, $\mathcal{O}(\frac{1}{\epsilon^2})$, while our algorithm has poly-logarithmic dependence in ϵ, $\mathcal{O}_d(\ln^d \frac{1}{\epsilon})$, where $\mathcal{O}_d$ hides an exponential constant in d. It remains open whether this $\ln^d \frac{1}{\epsilon}$ dependence in the oracle-complexity is necessary. To the best of our knowledge, this is the first example of an algorithm that improves over gradient descent while keeping its optimal memory usage in any regime where $\epsilon \leq \frac{1}{\sqrt{d}}$. While this improvement holds only in the exponential regime $\epsilon \leq \frac{1}{d^{\mathcal{O}(d)}}$, Theorem 8.1 still provides a non-trivial trade-off whenever $\ln \frac{1}{\epsilon} \gg \ln d$, and improves over the known memory-constrained center-of-mass in the standard regime $\epsilon \leq \frac{1}{\sqrt{d}}$ [WS19]. Fig. 8.1 depicts the trade-offs in the two regimes mentioned earlier.

Last, we note that the lower-bound trade-offs presented in [Mar+22] and in Chapter 7 do not show a dependence in the accuracy ϵ. Especially in the regime when $\ln \frac{1}{\epsilon} \gg \ln d$, this yields sub-optimal lower bounds (in fact even in the regime $\epsilon = 1/\text{poly}(d)$, our more careful analysis improves the lower bound on the memory by a $\ln d$ factor). We show with simple arguments that one can extend their results to include a $\ln \frac{1}{\epsilon}$ factor for both memory and query complexity. Fig. 8.1 presented these improved lower bounds.

Theorem 8.2. *For* $\epsilon \leq 1/\text{poly}(d)$ *and any* $\delta \in [0, 1]$ *(the notation* $\tilde{\Omega}$ *hides* $\ln^{\mathcal{O}(1)} d$ *factors),*

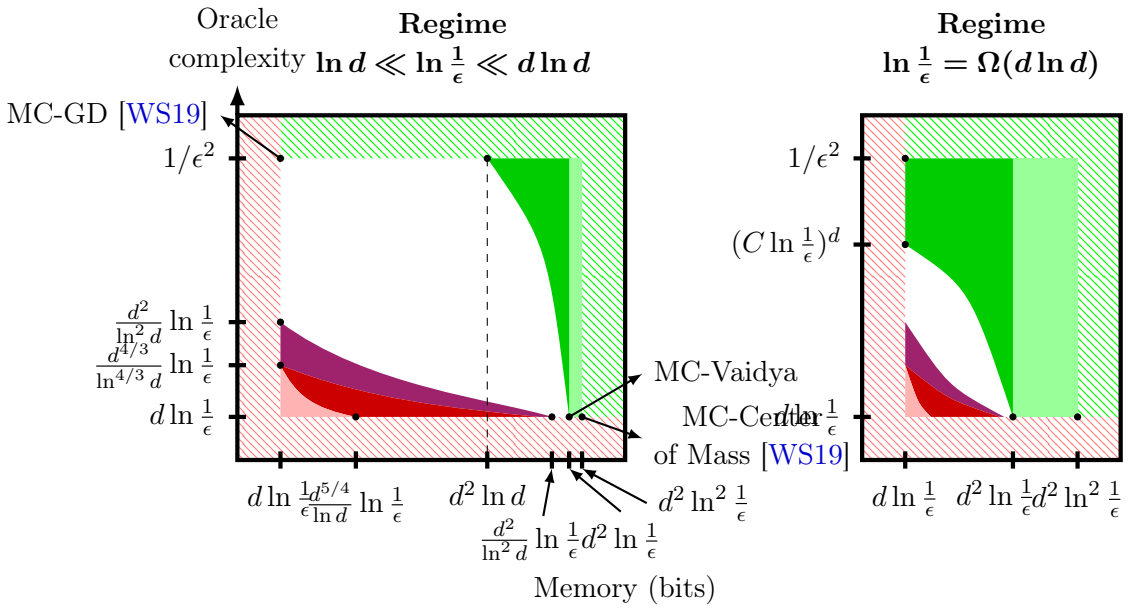

Figure 8.1: Trade-offs between available memory and first-order oracle-complexity for the feasibility problem over the unit ball. MC=Memory-constrained. GD=Gradient Descent. The left picture corresponds to the regime $\epsilon \gg d^{-\Omega(d)}$ and $\epsilon \leq 1/\text{poly}(d)$ and the right picture represents the regime $\epsilon \leq d^{-\mathcal{O}(d)}$. For both figures, the dashed pink "L" (resp. green inverted "L") region corresponds to historical lower (resp. upper) bounds for randomized algorithms. The solid pink (resp. red) lower bound trade-off is due to [Mar+22] (resp. Chapter 7) for randomized algorithms (resp. deterministic algorithms). The purple region is a lower bound trade-off for the feasibility problem for accuracy ϵ and deterministic algorithms we gave in Chapter 7. All these lower-bound trade-offs are represented with their $\ln \frac{1}{\epsilon}$ dependence (Theorem 8.2). We use memory-constrained Vaidya's method to gain a factor $\ln \frac{1}{\epsilon}$ in memory compared to memory-constrained center-of-mass [WS19], which gives the light green region, and a class of algorithms represented in dark green, that allows trading query-complexity for an extra $\ln \frac{1}{\epsilon} / \ln d$ factor saved in memory (Theorem 8.1). The dark green dashed region in the left figure emphasizes that the area covered by our class of algorithms depends highly on the regime for the accuracy ϵ: the resulting improvement in memory is more significant as ϵ is smaller. In the regime when $\epsilon \leq d^{-\mathcal{O}(d)}$ (right figure), our class of algorithms improves over the oracle-complexity of gradient descent while keeping the optimal memory $\mathcal{O}(d \ln \frac{1}{\epsilon})$.

1. *any (randomized) algorithm guaranteed to minimize* 1*-Lipschitz convex functions over the unit ball with accuracy* ϵ *uses* $d^{5/4-\delta} \ln \frac{1}{\epsilon}$ *bits of memory or makes* $\tilde{\Omega}(d^{1+4\delta/3} \ln \frac{1}{\epsilon})$ *queries,*

2. *any deterministic algorithm guaranteed to minimize* 1*-Lipschitz convex functions over the unit ball with accuracy* ϵ *uses* $d^{2-\delta} \ln \frac{1}{\epsilon}$ *bits of memory or makes* $\tilde{\Omega}(d^{1+\delta/3} \ln \frac{1}{\epsilon})$ *queries,*

3. *any deterministic algorithm guaranteed to solve the feasibility problem over the unit ball with accuracy* ϵ *uses* $d^{2-\delta} \ln \frac{1}{\epsilon}$ *bits of memory or makes* $\tilde{\Omega}(d^{1+\delta} \ln \frac{1}{\epsilon})$ *queries.*

The proof is given in Section 8.6 and the arguments therein could readily be used to exhibit the $\ln \frac{1}{\epsilon}$ dependence of potential future works improving over these lower bounds trade-offs.

Sketch of proof. At a high level, the results from [Mar+22] and Chapter 7 both used a barrier term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty$ where $\boldsymbol{A}$ has $\Theta(d)$ rows: if an algorithm does not have enough memory, $\boldsymbol{A}$ cannot be fully stored which in turn incurs a sub-optimal oracle-complexity. To achieve a $\ln \frac{1}{\epsilon}$ improvement in memory (Section 8.6.1), we modify the sampling of rows of $\boldsymbol{A}$, from uniform on vertices of the hypercube to uniform in an ϵ-net. The proof can then be adapted accordingly. Last, one can improve the oracle-complexity by a $\ln \frac{1}{\epsilon} / \ln d$ factor (Section 8.6.2) using a standard rescaling argument [NYD83].

8.4 Feasibility Problem Without Computations

In this section, we present a class of algorithms that are memory-constrained according to Definition 8.1 and achieve the desired memory and oracle-complexity bounds. We emphasize that the memory constraint is only applied between calls to the oracle and as a result, the algorithm is allowed infinite computation memory and computation power between calls to the oracle.

We start by defining discretization functions that will be used in our algorithms. For $\xi > 0$ and $x \in [-1, 1]$, we pose $\mathsf{Discretize}_1(x, \xi) = sign(x) \cdot \xi \lfloor |x|/\xi \rfloor$. Next, we define the discretization $\mathsf{Discretize}_d$ for general dimensions $d \geq 1$. For any $\boldsymbol{x} \in C$ and $\xi > 0$,

$$\mathsf{Discretize}_d(\boldsymbol{x}, \xi) = \left(\mathsf{Discretize}_1\left(x_1, \xi/\sqrt{d}\right), \ldots, \mathsf{Discretize}_1\left(x_d, \xi/\sqrt{d}\right)\right).$$

One can easily check that for any $\boldsymbol{x} \in C$,

$$\|\boldsymbol{x} - \mathsf{Discretize}_d(\boldsymbol{x}, \xi)\| \leq \xi \qquad \text{and} \qquad \|\mathsf{Discretize}_d(\boldsymbol{x}, \xi)\| \leq \|\boldsymbol{x}\|. \tag{8.1}$$

Further, to represent any output of $\mathsf{Discretize}_d(\cdot, \xi)$, one needs at most $d \ln \frac{2\sqrt{d}}{\xi} = \mathcal{O}(d \ln \frac{d}{\xi})$ bits.

8.4.1 Memory-constrained Vaidya's method

Our algorithm recursively uses Vaidya's cutting-plane method [Vai96] and subsequent works expanding on this method. We briefly describe the method. Given a polyhedron $\mathcal{P} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$, we define $s_i(\boldsymbol{x}) = \boldsymbol{a}_i^\top \boldsymbol{x} - b_i$ and $\boldsymbol{S}_x = diag(s_i(x), i \in [d])$. We will also use the shorthand $\boldsymbol{A}_x = \boldsymbol{S}_x^{-1}\boldsymbol{A}$. The volumetric barrier is defined as

$$V_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x}) = \frac{1}{2} \ln \det(\boldsymbol{A}_x^\top \boldsymbol{A}_x).$$

At each step, Vaidya's method queries the volumetric center of the polyhedron, which is the point minimizing the volumetric barrier. For convenience, we denote by $\mathsf{VolumetricCenter}$

this function, i.e., for any $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^d$ defining a non-empty polyhedron $\mathcal{P} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$,

$$\mathsf{VolumetricCenter}(\boldsymbol{A}, \boldsymbol{b}) = \arg \min_{\boldsymbol{x}: \boldsymbol{A}\boldsymbol{x} > \boldsymbol{b}} V_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x}).$$

When the polyhedron is unbounded, we can for instance take $\mathsf{VolumetricCenter}(\boldsymbol{A}, \boldsymbol{b}) = \boldsymbol{0}$. Vaidya's method makes use of leverage scores for each constraint i of the polyhedron, defined as $\sigma_i = (\boldsymbol{A}_x \boldsymbol{H}^{-1} \boldsymbol{A}_x^\top)_{i,i}$, where $\boldsymbol{H} = \boldsymbol{A}_x^\top \boldsymbol{A}_x$. We are now ready to define the update procedure for the polyhedron considered by Vaidya's volumetric method. We denote by $\mathcal{P}_t$ the polyhedron stored in memory after making t queries. The method keeps in memory the constraints defining the current polyhedron and the iteration index k when these constraints were added, which will be necessary for our next procedures. Hence, the polyhedron will be stored in the form $\mathcal{P}_t = \{(k_i, \boldsymbol{a}_i, b_i), i \in [m]\}$, and the associated constraints are given via $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$ where $\boldsymbol{A}^\top = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_m]$ and $\boldsymbol{b}^\top = [b_1, \dots, b_m]$. By abuse of notation, we will write $\mathsf{VolumetricCenter}(\mathcal{P})$ for the volumetric center of the polyhedron $\mathsf{VolumetricCenter}(\boldsymbol{A}, \boldsymbol{b})$ where $\boldsymbol{A}$ and $\boldsymbol{b}$ define the constraints stored in $\mathcal{P}$.

Initially, the polyhedron is simply $\mathcal{C}_d$, these constraints are given -1 index for convenience, and they will not play a role in the next steps. At each iteration, if the constraint $i \in [m]$ with minimum leverage score σ_i falls below a given threshold σ_{min}, it is removed from the polyhedron. Otherwise, we query the volumetric center of the current polyhedron and add the separation hyperplane as a constraint to the polyhedron. We bound the number of iterations of the procedure by

$$T(\delta, d) = \left\lceil c \cdot d \left(1.4 \ln \frac{1}{\delta} + 2 \ln d + 2 \ln(1 + 1/\sigma_{min}) \right) \right\rceil,$$

where σ_{min} and c are parameters that will be fixed shortly. Instead of making a call directly to the oracle O_S, we instead suppose that one has access to an oracle $O : \mathcal{I}_d \to \mathbb{R}^d$ where $\mathcal{I}_d = (\mathbb{Z} \times \mathbb{R}^{d+1})^\star$ has exactly the shape of the memory storing the information from the polyhedron. This form of oracle is used in our recursive calls to Vaidya's method. For example, such an oracle can simply be $O : \mathcal{P} \in \mathcal{I}_d \mapsto O_S(\mathsf{VolumetricCenter}(\mathcal{P}))$. Last, in our recursive method, we will not assume that oracle responses are normalized. As a result, we specify that if the norm of the response is too small, we can stop the algorithm. We assume however that the oracle already returns discretized vectors, which will be ensured in the following procedures. The cutting-plane algorithm is formally described in Algorithm 8.1. With an appropriate choice of parameters, this procedure finds an approximate solution of feasibility problems. We base the constants from [Ans98].

Lemma 8.1. *Fix $\sigma_{min} = 0.04$ and $c = \frac{1}{0.0014} \approx 715$. Let $\delta, \xi \in (0, 1)$ and $O : \mathcal{I}_d \to \mathbb{R}^d$. Write $\mathcal{P} = \{(k_i, \boldsymbol{a}_i, b_i), i \in [m]\}$ as the output of Algorithm 8.1 run with O, δ and ξ. Then,*

$$\min_{\substack{\lambda_i \geq 0, \ i \in [m], \\ \sum_{i \in [m]} \lambda_i = 1}} \max_{\boldsymbol{y} \in \mathcal{C}_d} \sum_{i=1}^m \lambda_i (\boldsymbol{a}_i^\top \boldsymbol{y} - b_i) = \max_{\boldsymbol{x} \in \mathcal{C}_d} \min_{i \in [m]} (\boldsymbol{a}_i^\top \boldsymbol{x} - b_i) \leq \delta.$$

Proof We first consider the case when the algorithm terminates because of a query $\boldsymbol{g} = O(\mathcal{P}_t)$ such that $\|\boldsymbol{g}\| \leq \delta/(2\sqrt{d})$. Then, for any $\boldsymbol{x} \in \mathcal{C}_d$, one directly has

$$\boldsymbol{g}^\top \boldsymbol{x} - b \leq \boldsymbol{g}^\top (\boldsymbol{x} - \boldsymbol{\omega}) \leq 2\sqrt{d} \|\boldsymbol{g}\| \leq \delta.$$

Input: $O : \mathcal{I}_d \to \mathbb{R}^d$, δ, $\xi \in (0,1)$

Let $T_{max} = T(\delta, d)$ and initialize $\mathcal{P}_0 := \{(-1, \boldsymbol{e}_i, -1), (-1, -\boldsymbol{e}_i, -1),\ i \in [d]\}$
for $t = 0, \dots, T_{max}$ **do**
  if $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\} = \emptyset$ **then return** $\mathcal{P}_t$;
  if $\min_{i \in [m]} \sigma_i < \sigma_{min}$ **then**
    $\mathcal{P}_{t+1} = \mathcal{P}_t \setminus \{(k_j, \boldsymbol{a}_j, b_j)\}$ where $j \in \arg\min_{i \in [m]} \sigma_i$
  else if $\boldsymbol{\omega} := \mathsf{VolumetricCenter}(\mathcal{P}_t) \notin \mathcal{C}_d$ **then**
    $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{(-1, -sign(\omega_j)\boldsymbol{e}_j, -1)\}$ where $j \in [d]$ has $|\omega_j| > 1$
  else
    $\boldsymbol{g} = O(\mathcal{P}_t)$ and $b = \xi \left\lceil \frac{\boldsymbol{g}^\top \boldsymbol{\omega}}{\xi} \right\rceil$, where $\boldsymbol{\omega} = \mathsf{VolumetricCenter}(\mathcal{P}_t)$
    $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{(t, \boldsymbol{g}, b)\}$
    if $\|\boldsymbol{g}\| \leq \delta$ **then return** $\mathcal{P}_{t+1}$;
end
return $\mathcal{P}_{T_{max}+1}$.

Algorithm 8.1: Memory-constrained Vaidya's volumetric method

where $\boldsymbol{\omega}$ is the volumetric center of the resulting polyhedron. In the second inequality we used the fact that $\boldsymbol{\omega} \in \mathcal{C}_d$, otherwise the algorithm would not have terminated at that step.

We next turn to the other cases and start by showing that the output polyhedron does not contain a ball of radius δ. This is immediate if the algorithm terminated because the polyhedron was empty. We then suppose this was not the case, and follow the same proof as given in [Ans98]. Algorithm 8.1 and the one provided in [Ans98] coincide when removing a constraint of the polyhedron. Hence, it suffices to consider the case when we add a constraint. We use the notation $\tilde{\boldsymbol{A}}^\top = [\boldsymbol{A}^\top, \boldsymbol{a}_{m+1}^\top]$, $\tilde{\boldsymbol{b}}^\top = [\boldsymbol{b}^\top, b_{m+1}]$ for the updated matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ after adding the constraint. We also denote $\boldsymbol{\omega} = \mathsf{VolumetricCenter}(\boldsymbol{A}, \boldsymbol{b})$ (resp. $\tilde{\boldsymbol{\omega}} = \mathsf{VolumetricCenter}(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}})$) the volumetric center of the polyhedron before (resp. after) adding the constraint. Next, we consider the vector $(\boldsymbol{b}')^\top = [\boldsymbol{b}^\top, \boldsymbol{a}_{m+1}^\top \boldsymbol{\omega}]$, which would have been obtained if the cut was performed at $\boldsymbol{\omega}$ exactly. We then denote $\boldsymbol{\omega}' = \mathsf{VolumetricCenter}(\tilde{\boldsymbol{A}}, \boldsymbol{b}')$. Then proof of [Ans98] shows that

$$V_{\tilde{\boldsymbol{A}}, \boldsymbol{b}'}(\boldsymbol{\omega}') \geq V_{\boldsymbol{A}, \boldsymbol{b}}(\boldsymbol{\omega}) + 0.0340.$$

We now observe that by construction, we have $\tilde{\boldsymbol{b}}_{m+1} \geq \boldsymbol{a}_{m+1}^\top \boldsymbol{\omega}$, so that the polyhedron associated to $(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}})$ is more constrained than the one associated to $(\tilde{\boldsymbol{A}}, \boldsymbol{b}')$. As a result, we have $V_{\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}}(\boldsymbol{x}) \geq V_{\tilde{\boldsymbol{A}}, \boldsymbol{b}'}(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathbb{R}^d$ such that $\tilde{\boldsymbol{A}}\boldsymbol{x} \geq \tilde{\boldsymbol{b}}$. Therefore,

$$V_{\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{\omega}}) \geq V_{\tilde{\boldsymbol{A}}, \boldsymbol{b}'}(\tilde{\boldsymbol{\omega}}) \geq V_{\tilde{\boldsymbol{A}}, \boldsymbol{b}'}(\boldsymbol{\omega}') \geq V_{\boldsymbol{A}, \boldsymbol{b}}(\boldsymbol{\omega}) + 0.0340.$$

This ends the modifications in the proof of [Ans98]. With the notations of this work, we still have $\Delta V^+ = 0.340$ and $\Delta V^- = 0.326$, so that $\Delta V = 0.0014$. Then, because $c = \frac{1}{\Delta V}$, the same proof shows that the procedure is successful for precision δ: the final polyhedron $(\boldsymbol{A}, \boldsymbol{b})$ returned by Algorithm 8.1 does not contains a ball of radius $> \delta$. As a result, whether the algorithm performed all T_{max} iterations or not, $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$ does not contain a ball of radius $> \delta'$, where $\boldsymbol{A}$ and $\boldsymbol{b}$ define the constraints stored in the output $\mathcal{P}$. Now letting m be

the objective value of the right optimization problem, there exists $\boldsymbol{x} \in \mathcal{C}_d$ such that for all $t \le T$, $\boldsymbol{g}_t^\top(\boldsymbol{x} - \boldsymbol{c}_t) \ge m$. Therefore, for any $\boldsymbol{x}' \in B_d(\boldsymbol{x}, m)$ one has

$$\forall i \in [m], \boldsymbol{a}_i^\top \boldsymbol{x}' - b_i \ge m + \boldsymbol{a}_t^\top(\boldsymbol{x}' - \boldsymbol{x}) \ge m - \|\boldsymbol{x}' - \boldsymbol{x}\| \ge 0.$$

In the last inequality we used $\|\boldsymbol{a}_t\| \le 1$. This implies that the polyhedron contains $B_d(\boldsymbol{x}, m)$. Hence, $m \le \delta$.

This ends the proof of the right inequality. The left equality is a direct application of strong duality for linear programming. ■

From now, we use the parameters $\sigma_{min} = 0.04$ and $c = 1/0.0014$ as in Lemma 8.1. Since the memory of both Vaidya's method and center-of-mass consists primarily of the constraints, we recall an important feature of Vaidya's method that the number of constraints at any time is $\mathcal{O}(d)$.

Lemma 8.2 ([Vai96; Ans97; Ans98])**.** *At any time while running Algorithm 8.1, the number of constraints of the current polyhedron is at most* $\frac{d}{\sigma_{min}} + 1$.

8.4.2 A recursive algorithm

We write $\mathcal{C}_{m+n} = \mathcal{C}_m \times \mathcal{C}_n$ and aim to apply Vaidya's method to the first m coordinates. To do so, we need to approximate a separation oracle on these m coordinates only, which corresponds to giving separation hyperplanes with small values for the last n coordinates. This can be achieved using the following auxiliary linear program. For $\mathcal{P} \in \mathcal{I}_n$, we define

$$\min_{\substack{\lambda_i \ge 0,\ i \in [m], \\ \sum_{i \in [m]} \lambda_i = 1}} \max_{\boldsymbol{y} \in \mathcal{C}_n} \sum_{i=1}^{m} \lambda_i(\boldsymbol{a}_i^\top \boldsymbol{y} - b_i), \quad m = |\mathcal{P}| \qquad (\mathcal{P}_{aux}(\mathcal{P}))$$

where as before, $\boldsymbol{A}$ and $\boldsymbol{b}$ define the constraints stored in $\mathcal{P}$. The procedure to obtain an approximate separation oracle on the first n coordinates $\mathcal{C}_n$ is given in Algorithm 8.2 and using Lemma 8.1 we can show that this procedure provides approximate separation vectors for the first n coordinates.

The next step involves using this approximation recursively. We write $d = \sum_{i=1}^{p} k_i$, and interpret $\mathcal{C}_d$ as $\mathcal{C}_{k_1} \times \cdots \times \mathcal{C}_{k_p}$. In particular, for $\boldsymbol{x} \in \mathcal{C}_d$, we write $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$ where $\boldsymbol{x}_i \in \mathcal{C}_{k_i}$ for $i \in [p]$. Applying Algorithm 8.2 recursively, we can obtain an approximate separation oracle for the first i coordinates $\mathcal{C}_{k_1} \times \cdots \times \mathcal{C}_{k_i}$. However, storing such separation vectors would be too memory-expensive, e.g., for $i = p$, that would correspond to storing the separation hyperplanes from the oracle O_S directly. Instead, given $j \in [i]$, Algorithm 8.3 recursively computes the $\boldsymbol{x}_j$ component of an approximate separation oracle for the first i variables $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i)$, via the procedure $\mathsf{ApproxOracle}(i, j)$.

We can then use $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(1, 1, \cdot)$ to solve the original problem with the memory-constrained Vaidya's method. In Section 8.4.3, we show that taking $\delta = \frac{\epsilon}{4d}$ and $\xi = \frac{\sigma_{min}\epsilon}{32d^{5/2}}$ achieves the desired oracle-complexity and memory usage. The final algorithm is given in Algorithm 8.4.

Input: δ, ξ, $O_x : \mathcal{I}_n \to \mathbb{R}^m$ and $O_y : \mathcal{I}_n \to \mathbb{R}^n$

Run Algorithm 8.1 with δ, ξ and O_y to obtain polyhedron $\mathcal{P}^\star$
Solve $\mathcal{P}_{aux}(\mathcal{P}^\star)$ to get a solution $\boldsymbol{\lambda}^\star$
Store $\boldsymbol{k}^\star = (k_i, i \in [m])$ where $m = |\mathcal{P}^\star|$, and $\boldsymbol{\lambda}^\star \leftarrow \mathsf{Discretize}(\boldsymbol{\lambda}^\star, \xi)$
Initialize $\mathcal{P}_0 := \{(-1, \boldsymbol{e}_i, -1), (-1 - \boldsymbol{e}_i, -1),\ i \in [d]\}$ and $\boldsymbol{u} = \mathbf{0} \in \mathbb{R}^m$
for $t = 0, 1, \ldots, \max_i k_i$ **do**
    if $t = k_i^\star$ *for some* $i \in [m]$ **then**
        $\boldsymbol{g}_x = O_x(\mathcal{P}_t)$
        $\boldsymbol{u} \leftarrow \mathsf{Discretize}_m(\boldsymbol{u} + \lambda_i^\star \boldsymbol{g}_x, \xi)$
    Update $\mathcal{P}_t$ to get $\mathcal{P}_{t+1}$ as in Algorithm 8.1
end
return $\boldsymbol{u}$

Algorithm 8.2: $\mathsf{ApproxSeparationVector}_{\delta,\xi}(O_x, O_y)$

Input: δ, ξ, $1 \le j \le i \le p$, $\mathcal{P}^{(r)} \in \mathcal{I}_{k_r}$ for $r \in [i]$, $O_S : \mathcal{C}_d \to \mathbb{R}^d$
if $i = p$ **then**
    $\boldsymbol{x}_r = \mathsf{VolumetricCenter}(\boldsymbol{A}_r, \boldsymbol{b}_r)$ where $(\boldsymbol{A}_r, \boldsymbol{b}_r)$ defines the constraints stored in $\mathcal{P}^{(r)}$ for $r \in [p]$
    $(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_p) = O_S(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$
    return $\mathsf{Discretize}_{k_j}(\boldsymbol{g}_j, \xi)$
end
Define $O_x : \mathcal{I}_{k_{i+1}} \to \mathbb{R}^{k_j}$ as $\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i+1, j, \mathcal{P}^{(1)}, , \ldots, \mathcal{P}^{(i)}, \cdot)$
Define $O_y : \mathcal{I}_{k_{i+1}} \to \mathbb{R}^{k_{i+1}}$ as $\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i+1, i+1, \mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(i)}, \cdot)$
return $\mathsf{ApproxSeparationVector}_{\delta,\xi}(O_x, O_y)$

Algorithm 8.3: $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i, j, \mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(i)})$

A geometric illustration of the recursive step. In Figure 8.2, we give a 2-dimensional feasibility problem with target $\boldsymbol{p}^* = (p_1^*, p_2^*)$ and two blocks (i.e. $p = 2$) as an illustration of our recursive approach (Algorithm 8.2) to construct an approximate separating hyperplane for a "reduced" problem.

Suppose at a step of the Algorithm 8.4, the current value of the x_1 coordinate is c. We aim to find an approximate separating hyperplane between $x_1 = p_1^*$ and $x_1 = c$. Algorithm 8.2 first runs Algorithm 8.1 (i.e. the memory-constrained Vaidya) to find two separating hyperplanes (the two blue hyperplanes). Lemma 8.1 then guarantees the existence of a convex combination of the 2 blue hyperplanes – the red hyperplane– which is approximately parallel to the x_2-axis and thus can serve as an approximate separating hyperplane between

Input: δ, ξ, and $\mathcal{O}_S : \mathcal{C}_d \to \mathbb{R}^d$ a separation oracle
Check: Throughout the algorithm, if O_S returned Success to a query $\boldsymbol{x}$, **return** $\boldsymbol{x}$
Run Algorithm 8.1 with parameters δ and ξ and oracle $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(1, 1, \cdot)$

Algorithm 8.4: Memory-constrained algorithm for convex optimization

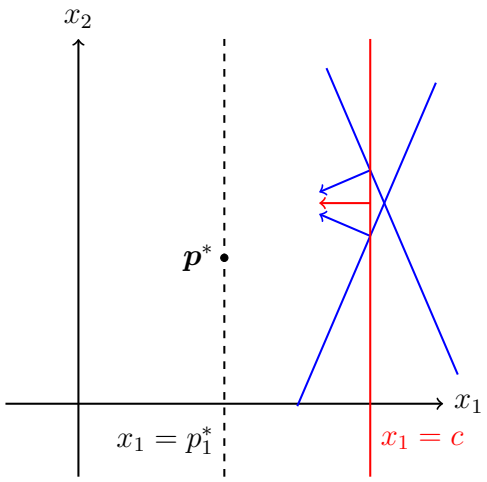

Figure 8.2: Intuition for the recursive procedure in Algorithm 8.4. Using the separation hyperplanes (blue) found by Algorithm 8.1, i.e., the memory-constrained Vaidya, it constructs an approximate separation hyperplane (red) between $x_1 = c$ and the target $x_1 = p_1^*$.

$x_1 = p_1^*$ and $x_1 = c$.

Intuitions on Algorithm 8.4 At the high level, the algorithm recursively runs Vaidya's method Algorithm 8.1 for each level of computation $i \in [p]$. Since each run of Algorithm 8.4 requires $\mathcal{O}(\frac{d}{p} \ln \frac{1}{\epsilon})$ queries, the total number of calls to the oracle, which is exponential in the number of levels, is $\mathcal{O}(\mathcal{O}(\frac{d}{p} \ln \frac{1}{\epsilon})^p)$. As for the memory usage, the algorithm mainly needs to keep in memory the constraints defining the polyhedrons at each level $i \in [p]$. From Lemma 8.2, each polyhedron only requires $\mathcal{O}(\frac{d}{p})$ constraints that each require $\mathcal{O}(\frac{d}{p} \ln \frac{1}{\epsilon})$ bits of memory. Hence, the total memory needed is $\mathcal{O}(\frac{d^2}{p} \ln \frac{1}{\epsilon})$. The main difficulty lies in showing that the algorithm is successful. To do so, we need to show that the precision in the successive approximated separation oracles Algorithm 8.2 is sufficient. To avoid an exponential dependence of the approximation error in p—which would be prohibitive for the memory usage of our method—each run of Vaidya's method Algorithm 8.1 is run for more iterations than the precision of the separation vectors would classically allow. To give intuition, if the separation oracle came from a convex optimization subgradient oracle for a function f, the iterates at a level i do not converge to the true "minimizer" of $\min_{\boldsymbol{x}_i} f^{(i)}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i)$, where $f^{(i)}(\cdot) = \min_{\boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_p} f(\cdot, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_p)$, but instead converge to a close enough point while still providing meaningful approximate subgradients at the higher level $i-1$ (in Algorithm 8.2).

8.4.3 Proof of the oracle-complexity and memory usage of Algorithm 8.4 without computation concerns

We first describe the recursive calls of Algorithm 8.3 in more detail. To do so, consider running the procedure $\mathsf{ApproxOracle}(i, j, \mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(i)})$ where $i < p$, which corresponds to running Algorithm 8.2 for specific oracles. We say that this is a level-i run. Then, the

algorithm performs at most $2T(\delta, k_{i+1})$ calls to $\mathsf{ApproxOracle}(i+1, i+1, \mathcal{P}^{(1)}, \dots, \mathcal{P}^{(i)}, \cdot)$, where the factor 2 comes from the fact that Vaidya's method Algorithm 8.1 is effectively run twice in Algorithm 8.2. The solution to $(\mathcal{P}_{aux}(\mathcal{P}))$ has as many components as constraints in the last polyhedron, which is at most $\frac{k_{i+1}}{\sigma_{min}} + 1$ by Lemma 8.2. Hence, the number of calls to $\mathsf{ApproxOracle}(i+1, j, \mathcal{P}^{(1)}, \dots, \mathcal{P}^{(i)}, \cdot)$ is at most $\frac{k_{i+1}}{\sigma_{min}} + 1$. In total, that is $\mathcal{O}(k_{i+1} \ln \frac{1}{\delta})$ calls to the level $i+1$ of the recursion.

We next aim to understand the output of running $\mathsf{ApproxOracle}(1, 1, \mathcal{P}^{(1)})$. We denote by $\boldsymbol{\lambda}(\mathcal{P}^{(1)})$ the solution $\mathcal{P}_{aux}(\mathcal{P}^\star)$ computed at l.2 of the first call to Algorithm 8.2, where $\mathcal{P}^\star$ is the output polyhedron of the first call to Algorithm 8.1. Denote by $\mathcal{S}(\mathcal{P}^{(1)})$ the set of indices of coordinates from $\boldsymbol{\lambda}(\mathcal{P}^{(1)})$ for which the procedure performed a call to $\mathsf{ApproxOracle}(2, 1, \mathcal{P}^{(1)}, \cdot)$. In other words, $\mathcal{S}(\mathcal{P}^{(1)})$ contains the indices of all coordinates of $\boldsymbol{\lambda}(\mathcal{P}^{(1)})$, except those for which the corresponding query lay outside of the unit cube, or the initial constraints of the cube. For any index $l \in \mathcal{S}(\mathcal{P}^{(1)})$, let $\mathcal{P}_l^{(2)}$ denote the state of the current polyhedron ($\mathcal{P}_t$ in l.7 of Algorithm 8.2) when that call was performed. Up to discretization issues, the output of the complete procedure is

$$\sum_{l \in \mathcal{S}(\mathcal{P}^{(1)})} \lambda_l(\mathcal{P}^{(1)}) \mathsf{ApproxOracle}(2, 1, \mathcal{P}^{(1)}, \mathcal{P}_l^{(2)}).$$

We continue in the recursion, defining $\boldsymbol{\lambda}(\mathcal{P}^{(1)}, \mathcal{P}_l^{(2)})$ and $\mathcal{S}(\mathcal{P}^{(1)}, \mathcal{P}_l^{(2)})$ for all $l \in \mathcal{S}(\mathcal{P}^{(1)})$, until we have defined all vectors of the form $\boldsymbol{\lambda}(\mathcal{P}^{(1)}, \mathcal{P}_{l_2}^{(2)}, \dots, \mathcal{P}_{l_r}^{(r)})$ and all sets of the form $\mathcal{S}(\mathcal{P}^{(1)}, \mathcal{P}_{l_2}^{(2)}, \dots, \mathcal{P}_{l_r}^{(r)})$ for $i+1 \le r \le p-1$. To simplify the notation and emphasize that all these polyhedra depend on the recursive computation path, we adopt the notation

$$\begin{aligned} \lambda^{l_2,\dots,l_{r+1}} &:= \lambda_{l_{r+1}}(\mathcal{P}^{(1)}, \mathcal{P}_{l_2}^{(2)}, \dots, \mathcal{P}_{l_r}^{(r)}) \\ \mathcal{S}^{l_2,\dots,l_r} &:= \mathcal{S}(\mathcal{P}^{(1)}, \mathcal{P}_{l_2}^{(2)}, \dots, \mathcal{P}_{l_r}^{(r)}) \end{aligned}$$

We recall that these polyhedron are kept in memory to query their volumetric center. For ease of notation, we write $\boldsymbol{x}_1 = \mathsf{VolumetricCenter}(\mathcal{P}^{(1)})$, and write $\boldsymbol{c}^{l_2,\dots,l_r} = \mathsf{VolumetricCenter}(\mathcal{P}_{l_r}^{(r)})$ for $2 \le r \le p$, where $l_2, \dots, l_{r-1}$ were the indices from the computation path leading up to $\mathcal{P}_{l_r}^{(r)}$. Last, we write $O_S = (O_{S,1}, \dots, O_{S,p})$, where $O_{S,i} : \mathcal{C}_d \to \mathbb{R}^{k_i}$ is the "$\boldsymbol{x}_i$" component of O_S, for all $i \in [p]$.

With these notations, we show that the output of $\mathsf{ApproxOracle}(i, j, \mathcal{P}^{(1)}, \mathcal{P}_{l_2}^{(2)}, \dots, \mathcal{P}_{l_i}^{(i)})$ is approximately equal to the vector

$$\begin{aligned} &G(i, j, \boldsymbol{x}_1, \boldsymbol{c}^{l_2}, \dots, \boldsymbol{c}^{l_2,\dots,l_i}) \\ &\qquad := \sum_{\substack{l_{i+1} \in \mathcal{S},\ l_{i+2} \in \mathcal{S}^{l_{i+1}}, \\ \dots\ ,\ l_p \in \mathcal{S}^{l_{i+1},\dots,l_{p-1}}}} \lambda^{l_{i+1}} \lambda^{l_{i+1},l_{i+2}} \cdots \lambda^{l_{i+1},\dots,l_p} \cdot O_{S,j}(\boldsymbol{x}_1, \boldsymbol{c}^{l_2}, \dots, \boldsymbol{c}^{l_2,\dots,l_p}), \end{aligned}$$

with the convention that for $i = p$,

$$G(p, j, \boldsymbol{x}_1, \boldsymbol{c}^{l_2}, \dots, \boldsymbol{c}^{l_2,\dots,l_p}) := O_{S,j}(\boldsymbol{x}_1, \boldsymbol{c}^{l_2}, \dots, \boldsymbol{c}^{l_2,\dots,l_p}).$$

The corresponding computation tree is represented in Fig. 8.3. For convenience, we omitted the term $j = 1$.

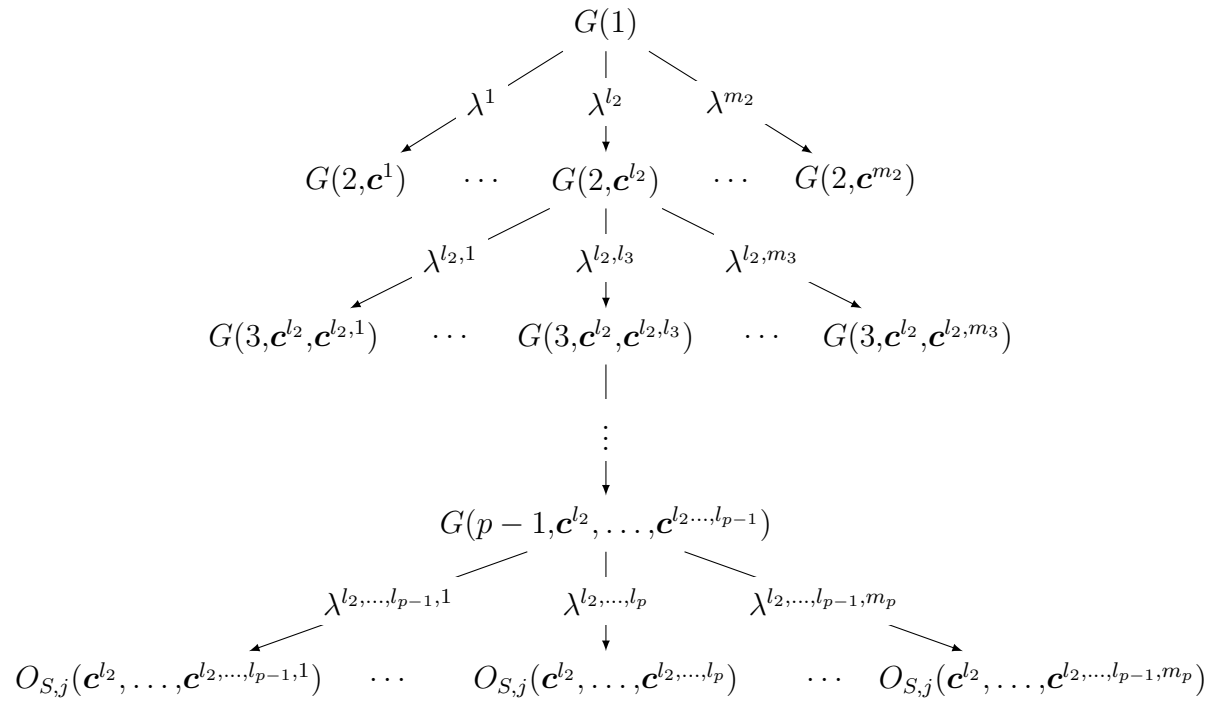

Figure 8.3: Computation tree representing the recursive calls to ApproxOracle starting from the calls to ApproxOracle$(1,1,\cdot)$ from Algorithm 8.4

We start the analysis with a simple result showing that if the oracle O_S returns separation vectors of norm bounded by one, then the responses from ApproxOracle also lie in the unit ball.

Lemma 8.3. *Fix $\delta,\xi\in(0,1)$, $1\leq j\leq i\leq p$ and an oracle $O_S=(O_{S,1},\ldots,O_{S,p}):\mathcal{C}_d\to\mathbb{R}^d$. Suppose that O_S takes values in the unit ball. For any $s\in[i]$ let $\mathcal{P}_{l_s}^{(s)}\in\mathcal{I}_{k_s}$ represent a bounded polyhedrons with $\mathsf{VolumetricCenter}(\mathcal{P}_{l_s}^{(s)})\in\mathcal{C}_{k_s}$. Then, one has*

$$\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)})\|\leq 1.$$

Proof We prove this by simple induction on i. For convenience, we define the point $\boldsymbol{x}_k=\mathsf{VolumetricCenter}(\mathcal{P}_{l_k}^{(k)})$. If $i=p$, we have

$$\begin{aligned}\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)})\| &= \|\mathsf{Discretize}_{k_j}(O_{S,j}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_p),\xi)\|\\ &\leq \|O_{S,j}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_p)\|\leq 1,\end{aligned}$$

where in the first inequality we used Eq (8.1) and in the second inequality we used the fact that $O_S(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_p)$ has norm at most one. Now suppose that the result holds for $i+1\leq p$. Then by construction, the output $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)})$ is the result of iterative discretizations. Using Eq (8.1) and the previously defined notations, we

obtain

$$\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)})\| \\ \leq \left\| \sum_{l_{i+1}\in\mathcal{S}^{l_1,\ldots,l_i}} \lambda^{l_2,\ldots,l_i}\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i+1,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)},\mathcal{P}_{l_{i+1}}^{(i+1)}) \right\| \leq 1.$$

In the last inequality, we used the induction hypothesis together with $\sum_{l_{i+1}} \lambda^{l_2,\ldots,l_{i+1}} \leq 1$ using Eq (8.1). This ends the induction and the proof. ■

We are now ready to compare the output of Algorithm 8.3 to $G(i,j,\boldsymbol{x}_1,\boldsymbol{c}^{l_2},\ldots,\boldsymbol{c}^{l_2,\ldots,l_i})$.

Lemma 8.4. *Fix $\delta,\xi\in(0,1)$, $1\leq j\leq i\leq p$ and an oracle $O_S=(O_{S,1},\ldots,O_{S,p}):\mathcal{C}_d\to\mathbb{R}^d$. Suppose that O_S takes values in the unit ball. For any $s\in[i]$ let $\mathcal{P}_{l_s}^{(s)}\in\mathcal{I}_{k_s}$ represent a bounded polyhedron with $\mathsf{VolumetricCenter}(\mathcal{P}_{l_s}^{(s)})\in\mathcal{C}_{k_s}$. Denote $\boldsymbol{x}_r=\boldsymbol{c}(\mathcal{P}_{l_r}^{(r)})$ for $r\in[i]$. Then,*

$$\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)}) - G(i,j,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i)\| \leq \frac{4}{\sigma_{min}}d\xi.$$

Proof We prove by simple induction on i that

$$\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_i}^{(i)}) - G(i,j,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i)\| \\ \leq \left(1+\frac{2}{\sigma_{min}}(k_{i+1}+\ldots+k_p)+2(p-i)\right)\xi.$$

First, for $i=p$, the result is immediate since the discretization is with precision ξ (l.4 of Algorithm 8.3). Now suppose that this is the case for $i\leq p$ and any valid values of other parameters. For conciseness, we write $\boldsymbol{G}=(\mathcal{P}_{l_1}^{(1)},\ldots,\mathcal{P}_{l_{i-1}}^{(i-1)})$. Next, recall that by Lemma 8.2, $|\mathcal{S}^{l_2,\ldots,l_{i-1}}|\leq\frac{k_i}{\sigma_{min}}+1$. Hence, the discretizations due to l.8 of Algorithm 8.2 can affect the estimate for at most that number of rounds. Then, we have

$$\left\|\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i-1,j,\boldsymbol{G}) - \sum_{l_i\in\mathcal{S}^{l_2,\ldots,l_{i-1}}} \tilde{\lambda}^{l_2,\ldots,l_i}\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\boldsymbol{G},\mathcal{P}_{l_i}^{(i)})\right\| \\ \leq \left(\frac{k_i}{\sigma_{min}}+1\right)\xi,$$

where $\tilde{\lambda}^{l_2,\ldots,l_i}$ are the discretized coefficients that are used during the computation l.8 of Algorithm 8.2. Now using Lemma 8.3, we have

$$\left\| \sum_{l_i\in\mathcal{S}^{l_2,\ldots,l_{i-1}}} (\tilde{\lambda}^{l_2,\ldots,l_i}-\lambda^{l_2,\ldots,l_i})\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,j,\boldsymbol{G},\mathcal{P}_{l_i}^{(i)}) \right\| \\ \leq \|\tilde{\boldsymbol{\lambda}}^{l_{i+1},\ldots,l_{i-1}}-\boldsymbol{\lambda}^{l_{i+1},\ldots,l_{i-1}}\|_1 \leq \left(\frac{k_i}{\sigma_{min}}+1\right)\xi.$$

In the last inequality we used the fact that $\boldsymbol{\lambda}$ has at most $\frac{k_i}{\sigma_{min}}+1$ non-zero coefficients. As a result, using the induction for each term of the sum, and the fact that $\sum_{l_i}\lambda^{l_2,\ldots,l_i}\leq 1$, we obtain

$$\begin{aligned}\|\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i-1,j,\boldsymbol{G})-G(i-1,j,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{i-1})\|\\ \leq\left(1+\frac{2}{\sigma_{min}}(k_{i+1}+\ldots+k_p)+2(p-i)\right)\xi+\left(\frac{2k_i}{\sigma_{min}}+2\right)\xi,\end{aligned}$$

which completes the induction. Noting that $k_{i+1}+\ldots+k_p\leq k_1+\ldots+k_p\leq d$ and $p-i\leq d-1$ ends the proof. ■

Next, we show that the outputs of Algorithm 8.3 provide approximate separation hyperplanes for the first i coordinates $(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i)$.

Lemma 8.5. *Fix $\delta,\xi\in(0,1)$, $1\leq j\leq i\leq p$ and an oracle $O_S=(O_{S,1},\ldots,O_{S,p}):\mathcal{C}_d\to\mathbb{R}^d$ for accuracy $\epsilon>0$. Suppose that O_S takes values in the unit ball $B_d(0,1)$. For any $s\in[i]$ let $\mathcal{P}^{(s)}_{l_s}\in\mathcal{I}_{k_s}$ represent a bounded polyhedron with $\mathsf{VolumetricCenter}(\mathcal{P}^{(s)}_{l_s})\in\mathcal{C}_{k_s}$. Denote $\boldsymbol{x}_r=\boldsymbol{c}(\mathcal{P}^{(r)}_{l_r})$ for $r\in[i]$. Suppose that when running $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,i,\mathcal{P}^{(1)}_{l_1},\ldots,\mathcal{P}^{(i)}_{l_i})$, no successful vector was queried. Then, any vector $\boldsymbol{x}^\star=(\boldsymbol{x}^\star_1,\ldots,\boldsymbol{x}^\star_p)\in\mathcal{C}_d$ such that $B_d(\boldsymbol{x}^\star,\epsilon)$ is contained in the successful set satisfies*

$$\sum_{r\in[i]}\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,r,\mathcal{P}^{(1)}_{l_1},\ldots,\mathcal{P}^{(i)}_{l_i})^\top(\boldsymbol{x}^\star_r-\boldsymbol{x}_r)\geq\epsilon-\frac{8d^{5/2}}{\sigma_{min}}\xi-d\delta.$$

Proof For $i\leq r\leq p$ and $j\leq r$, we use the notation

$$\boldsymbol{g}_j^{l_{i+1},\ldots,l_r}=\mathsf{ApproxOracle}_{\delta,\xi,O_S}(r,j,\mathcal{P}^{(1)}_{l_1},\ldots,\mathcal{P}^{(r)}_{l_r}).$$

Using Lemma 8.4, we always have for $j\in[r]$,

$$\|\boldsymbol{g}_j^{l_{i+1},\ldots,l_r}-G(r,j,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i,\boldsymbol{c}^{l_{i+1}},\ldots,\boldsymbol{c}^{l_{i+1},\ldots,l_r})\|\leq\frac{4d}{\sigma_{min}}\xi.\tag{8.2}$$

Also, observe that by Lemma 8.3 the recursive outputs of $\mathsf{ApproxOracle}$ always have norm bounded by one.

Next, let $\mathcal{T}^{l_{i+1},\ldots,l_{r-1}}$ be the set of indices corresponding to coordinates of $\boldsymbol{\lambda}^{l_{i+1},\ldots,l_{r-1}}$ for which the procedure $\mathsf{ApproxOracle}$ did not call for a level-r computation. These correspond to 1. constraints from the initial cube $\mathcal{P}_0$, or 2. cases when the volumetric center was out of the unit cube (l.6-7 of Algorithm 8.1) and as a result, the index of the added constraint was -1 instead of the current iteration index t. Similarly as above, for any $t\in\mathcal{T}^{l_{i+1},\ldots,l_{r-1}}$, we denote by $\boldsymbol{g}_r^{l_{i+1},\ldots,l_{r-1},t}$ the corresponding vector $\boldsymbol{a}_t$. We recall that by construction, this vector is of the form $\pm\boldsymbol{e}_j$ for some $j\in[k_r]$. Then, from Lemma 8.1, since the responses of the oracle always have norm bounded by one, for all $\boldsymbol{y}_r\in\mathcal{C}_{k_r}$,

$$\sum_{l_r\in\mathcal{S}^{l_{i+1},\ldots,l_{r-1}}\cup\mathcal{T}^{l_{i+1},\ldots,l_{r-1}}}\lambda^{l_{i+1},\ldots,l_r}(\boldsymbol{g}_r^{l_{i+1},\ldots,l_r})^\top(\boldsymbol{y}_r-\boldsymbol{c}^{l_{i+1},\ldots,l_r})\leq\delta.\tag{8.3}$$

For conciseness, we use the shorthand $(\mathcal{S} \cup \mathcal{T})^{l_{i+1},\dots,l_{r-1}} := \mathcal{S}^{l_{i+1},\dots,l_{r-1}} \cup \mathcal{T}^{l_{i+1},\dots,l_{r-1}}$, which contains all indices from coordinates of $\boldsymbol{\lambda}^{l_{i+1},\dots,l_{r-1}}$. In particular,

$$\sum_{l_r \in (\mathcal{S}\cup\mathcal{T})^{l_{i+1},\dots,l_{r-1}}} \lambda^{l_{i+1},\dots,l_r} = 1. \tag{8.4}$$

We now proceed to estimate the precision of the vectors $G(i, j, \boldsymbol{x}_1, \dots, \boldsymbol{x}_i)$ as approximate separation hyperplanes for coordinates $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i)$. Let $\boldsymbol{x}^\star \in \mathcal{C}_d$ such that $B_d(\boldsymbol{x}^\star, \epsilon)$ is within the successful set. Then, for any choice of $l_{i+1} \in \mathcal{S}, \dots, l_p \in \mathcal{S}^{l_{i+1},\dots,l_{p-1}}$, since we did not query a successful vector, we have for all $\boldsymbol{z} \in B_d(\boldsymbol{x}^\star, \epsilon)$,

$$O_S(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})^\top (\boldsymbol{z} - (\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})) \geq 0.$$

As a result, because the responses from O_S have unit norm,

$$O_S(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})^\top (\boldsymbol{x}^\star - (\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})) \geq \epsilon. \tag{8.5}$$

Now write $\boldsymbol{x}^\star = (\boldsymbol{x}^\star_1, \dots, \boldsymbol{x}^\star_p)$. In addition to the previous equation, for $l_{i+1} \in \mathcal{S}, \dots, l_{r-1} \in \mathcal{S}^{l_{i+1},\dots,l_{r-2}}$ and any $l_r \in \mathcal{T}^{l_{i+1},\dots,l_{r-1}}$, one has $(\boldsymbol{g}_r^{l_{i+1},\dots,l_r})^\top \boldsymbol{x}^\star_r + 1 \geq \epsilon$, because $\boldsymbol{x}^\star$ is within the cube $\mathcal{C}_d$ and at least at distance ϵ from the constraints of the cube. Similarly as when $l_r \in \mathcal{S}^{l_{i+1},\dots,l_{r-1}}$, for any $l_r \in \mathcal{T}^{l_{i+1},\dots,l_{r-1}}$ we denote by $\boldsymbol{c}^{l_{i+1},\dots,l_r}$ the volumetric center of the polyhedron $\mathcal{P}^{(r)}_{l_r}$ along the corresponding computation path, if l_r corresponded to an added constraints when $\boldsymbol{c}^{l_{i+1},\dots,l_r} \notin \mathcal{C}_{k_r}$. Otherwise, if l_r corresponded to the constraint $\boldsymbol{a} = \pm \boldsymbol{e}_j$ of the initial cube, we pose $\boldsymbol{c}^{l_{i+1},\dots,l_r} = -\boldsymbol{a}$. Now by construction, in both cases one has $(\boldsymbol{g}_r^{l_{i+1},\dots,l_r})^\top \boldsymbol{c}^{l_{i+1},\dots,l_r} \leq -1$ (l.7 of Algorithm 8.1). Thus,

$$(\boldsymbol{g}_r^{l_{i+1},\dots,l_r})^\top (\boldsymbol{x}^\star_r - \boldsymbol{c}^{l_{i+1},\dots,l_r}) \geq \epsilon. \tag{8.6}$$

Recalling Eq (8.4), we then sum all equations of the form Eq (8.5) and Eq (8.6) along the computation path, to obtain

$$\begin{aligned}(A) := \sum_{\substack{l_{i+1}\in\mathcal{S},\dots,\\ l_p\in\mathcal{S}^{l_{i+1},\dots,l_{p-1}}}} & \lambda^{l_{i+1}} \dots \lambda^{l_{i+1},\dots,l_p} \\ & \cdot O_S(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})^\top (\boldsymbol{x}^\star - (\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_p})) \\ + \sum_{i+1\leq r\leq p} \sum_{\substack{l_{i+1}\in\mathcal{S},\dots,l_{r-1}\in\mathcal{S}^{l_{i+1},\dots,l_{r-2}},\\ l_r\in\mathcal{T}^{l_{i+1},\dots,l_{r-1}}}} & \lambda^{l_{i+1}} \dots \lambda^{l_{i+1},\dots,l_r} \cdot (\boldsymbol{g}_r^{l_{i+1},\dots,l_r})^\top (\boldsymbol{x}^\star_r - \boldsymbol{c}^{l_{i+1},\dots,l_r}) \geq \epsilon.\end{aligned}$$

Now using the convention

$$G(r, r, \boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_r}) := \boldsymbol{g}_r^{l_{i+1},\dots,l_r}, \qquad l_r \in \mathcal{T}^{l_{i+1},\dots,l_{r-1}},$$

for any $l_{i+1} \in \mathcal{S}, \dots, l_{r-1} \in \mathcal{S}^{l_{i+1},\dots,l_{r-2}}$, we can write

$$\begin{aligned}(A) = \sum_{r\leq i} G(i, r, \boldsymbol{x}_1, \dots, \boldsymbol{x}_i)^\top (\boldsymbol{x}^\star_r - \boldsymbol{x}_r) + \sum_{i+1\leq r\leq p} \sum_{\substack{l_{i+1}\in\mathcal{S},\dots,\\ l_{r-1}\in\mathcal{S}^{l_{i+1},\dots,l_{r-2}}}} & \lambda^{l_{i+1}} \dots \lambda^{l_{i+1},\dots,l_{r-1}} \\ \times \sum_{l_r\in(\mathcal{S}\cup\mathcal{T})^{l_{i+1},\dots,l_{r-1}}} & \lambda^{l_{i+1},\dots,l_r} G(r, r, \boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \boldsymbol{c}^{l_{i+1}}, \dots, \boldsymbol{c}^{l_{i+1},\dots,l_r})^\top (\boldsymbol{x}^\star_r - \boldsymbol{c}^{l_{i+1},\dots,l_r}).\end{aligned}$$

We next relate the terms G to the output of ApproxOracle. For simplicity, let us write $\boldsymbol{G} = (\mathcal{P}_{l_1}^{(1)}, \dots, \mathcal{P}_{l_i}^{(i)})$, which by abuse of notation was assimilated to $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i)$. Recall that by construction and hypothesis, all points where the oracle was queried belong to $\mathcal{C}_d$, so that for instance $\|\boldsymbol{x}_r^\star - \boldsymbol{c}^{l_{i+1},\dots,l_r}\| \le 2\sqrt{k_r} \le 2\sqrt{d}$ for any $l_r \in \mathcal{S}^{l_{i+1},\dots,l_{r-1}}$. Using the above equations together with Eq (8.2) and Lemma 8.4 gives

$$\epsilon \le \sum_{r\le i}\left[\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i,r,\boldsymbol{G})^\top(\boldsymbol{x}_r^\star - \boldsymbol{x}_r) + \frac{8d^{3/2}}{\sigma_{min}}\xi\right] + \sum_{i+1\le r\le p}\sum_{\substack{l_{i+1}\in\mathcal{S},\dots,\\ l_{r-1}\in\mathcal{S}^{l_{i+1},\dots,l_{r-2}}}}$$
$$\lambda^{l_{i+1}}\cdots\lambda^{l_{i+1},\dots,l_{r-1}}\sum_{l_r\in(\mathcal{S}\cup\mathcal{T})^{l_{i+1},\dots,l_{r-1}}}\lambda^{l_{i+1},\dots,l_r}\left[(\boldsymbol{g}_r^{l_{i+1},\dots,l_r})^\top(\boldsymbol{x}_r^\star - \boldsymbol{c}^{l_{i+1},\dots,l_r}) + \frac{8d^{3/2}}{\sigma_{min}}\xi\right]$$
$$\le \frac{8pd^{3/2}}{\sigma_{min}}\xi + (p-i)\delta + \sum_{r\le i}\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i,r,\boldsymbol{G})^\top(\boldsymbol{x}_r^\star - \boldsymbol{x}_r)$$

where in the second inequality, we used Eq (8.3). Using $p \le d$, this ends the proof of the lemma. ■

We are now ready to show that Algorithm 8.4 is a valid algorithm for convex optimization.

Theorem 8.3. *Let $\epsilon \in (0,1)$ and $O_S : \mathcal{C}_d \to \mathbb{R}^d$ be a separation oracle such that the successful set contains a ball of radius ϵ. Pose $\delta = \frac{\epsilon}{4d}$ and $\xi = \frac{\sigma_{min}\epsilon}{32d^{5/2}}$. Next, let $p \ge 1$ and $k_1, \dots, k_p \le \lceil \frac{d}{p} \rceil$ such that $k_1 + \dots + k_p = d$. With these parameters, Algorithm 8.4 finds a successful vector with $(C\frac{d}{p}\ln\frac{d}{\epsilon})^p$ queries and using memory $\mathcal{O}(\frac{d^2}{p}\ln\frac{d}{\epsilon})$, for some universal constant $C > 0$.*

Proof Suppose by contradiction that Algorithm 8.4 never queried a successful point. Then, with the chosen parameters, Lemma 8.5 shows that, for any vector $\boldsymbol{x}^\star = (\boldsymbol{x}_1^\star, \dots, \boldsymbol{x}_p^\star)$ such that $B_d(\boldsymbol{x}^\star, \epsilon)$ is within the successful set, with the same notations, one has

$$\sum_{r\le i}\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i,r,\mathcal{P}_{l_1}^{(1)},\dots,\mathcal{P}_{l_i}^{(i)})^\top(\boldsymbol{x}_r^\star - \boldsymbol{x}_r) \ge \epsilon - \frac{8d^{5/2}}{\sigma_{min}}\xi - d\delta \ge \frac{\epsilon}{2}.$$

Now denote by $(\boldsymbol{a}_t, b_t)$ the constraints that were added at any time during the run of Algorithm 8.1 when using the oracle ApproxOracle with $i = j = 1$. The previous equation shows that for all such constraints,

$$\boldsymbol{a}_t^\top\boldsymbol{x}_1^\star - b_t \ge \boldsymbol{a}_t^\top(\boldsymbol{x}_1^\star - \omega_t) - \xi \ge \frac{\epsilon}{2} - \xi,$$

where $\boldsymbol{\omega}_t$ is the volumetric center of the polyhedron at time t during Vaidya's method Algorithm 8.1. Now, since the algorithm terminated, by Lemma 8.1, we have that

$$\min_t(\boldsymbol{a}_t^\top\boldsymbol{x}_1^\star - b_t) \le \delta.$$

This is absurd since $\delta + \xi < \frac{\epsilon}{2}$. This ends the proof that Algorithm 8.4 finds a successful vector.

We now estimate its oracle-complexity and memory usage. First, recall that a run of ApproxOracle of level i makes $\mathcal{O}(k_{i+1} \ln \frac{1}{\delta})$ calls to level-$(i+1)$ runs of ApproxOracle. As a result, the oracle-complexity $Q_d(\epsilon; k_1, \ldots, k_p)$ satisfies

$$Q_d(\epsilon; k_1, \ldots, k_p) = \left(Ck_1 \ln \frac{1}{\delta}\right) \times \ldots \times \left(Ck_p \ln \frac{1}{\delta}\right) \leq \left(C' \frac{d}{p} \log \frac{d}{\epsilon}\right)^p$$

for some universal constants $C, C' \geq 2$.

We now turn to the memory of the algorithm. For each level $i \in [p]$ of runs for ApproxOracle, we keep memory placements for

1. the value $j^{(i)}$ of the corresponding call to ApproxOracle$(i, j^{(i)}, \cdot)$ (for l.6-7 of Algorithm 8.3): $\mathcal{O}(\ln d)$ bits,
2. the iteration number $t^{(i)}$ during the run of Algorithm 8.1 or within Algorithm 8.2: $\mathcal{O}(\ln(k_i \ln \frac{1}{\delta}))$ bits
3. the polyhedron constraints contained in the state of $\mathcal{P}^{(i)}$: $\mathcal{O}(k_i \times k_i \ln \frac{1}{\xi})$ bits,
4. potentially, already computed dual variables $\boldsymbol{\lambda}^\star$ and their corresponding vector of constraint indices $\boldsymbol{k}^\star$ (l.3 of Algorithm 8.2): $\mathcal{O}(k_i \times \ln \frac{1}{\xi})$ bits,
5. the working vector $\boldsymbol{u}^{(i)}$ (updated l.8 of Algorithm 8.2): $\mathcal{O}(k_i \ln \frac{1}{\xi})$ bits.

The memory structure is summarized in Table 8.1.

We can then check that this memory is sufficient to run Algorithm 8.4. An important point is that for any run of ApproxOracle$(i, j, \cdot)$, in Algorithm 8.2, after running Vaidya's method Algorithm 8.1 and storing the dual variables $\boldsymbol{\lambda}^\star$ and corresponding indices $\boldsymbol{k}^\star$ within their placements $(\boldsymbol{k}^{\star(i)}, \boldsymbol{\lambda}^{\star(i)})$ (l.1-3 of Algorithm 8.2), the iteration index $t^{(i)}$ and polyhedron $\mathcal{P}^{(i)}$ memory placements are reset and can be used again for the second run of Vaidya's method (l.4-10 of Algorithm 8.2). During this second run, the vector $\boldsymbol{u}$ is stored in its corresponding memory placement $\boldsymbol{u}^{(i)}$ and updated along the algorithm. Once this run is finished, the output of ApproxOracle$(i, j, \cdot)$ is readily available in the placement $\boldsymbol{u}^{(i)}$. For $i = p$, the algorithm does not need to wait for the output of a level-$(i+1)$ computation and can directly use the $j^{(p)}$-th component of the returned separation vector from the oracle O_S. As a result, the number of bits of memory used throughout the algorithm is at most

$$M = \sum_{i=1}^{p} \mathcal{O}\left(k_i^2 \ln \frac{1}{\xi}\right) = \mathcal{O}\left(\frac{d^2}{p} \ln \frac{d}{\epsilon}\right).$$

This ends the proof of the theorem. ■

We can already give the useful range for p for our algorithms, which will also apply to the case with computational-memory constraints Section 8.5.

Proof of Corollary 8.1 Suppose $\epsilon \geq \frac{1}{d^d}$. Then, for some $p_{max} = \Theta(\frac{C \ln \frac{1}{\epsilon}}{2 \ln d}) \leq d$, the algorithm from Theorem 8.1 yields a $\mathcal{O}(\frac{1}{\epsilon^2})$ oracle-complexity. On the other hand, if $\epsilon \leq \frac{1}{d^d}$, we can take $p_{max} = d$, which gives an oracle-complexity $\mathcal{O}((C \ln \frac{1}{\epsilon})^d)$. ■

i	1	$\dots$	p
j	$j^{(1)}$		$j^{(p)}$
Iteration index	$t^{(1)}$		$t^{(p)}$
Polyhedron	$\mathcal{P}^{(1)} = \begin{pmatrix} k_1, \boldsymbol{a}_1, b_1 \\ k_2, \boldsymbol{a}_2, b_2 \\ \dots \\ k_m, \boldsymbol{a}_m, b_m \end{pmatrix}$		$\mathcal{P}^{(p)}$
Computed dual variables	$(\boldsymbol{k}^{\star(1)}, \boldsymbol{\lambda}^{\star(1)}) = \begin{pmatrix} k_1^\star, \lambda_1^\star \\ k_2^\star, \lambda_2^\star \\ \dots \end{pmatrix}$		$(\boldsymbol{k}^{\star(p)}, \boldsymbol{\lambda}^{\star(p)})$
Working separation vector	$\boldsymbol{u}^{(1)}$		$\boldsymbol{u}^{(p)}$

Table 8.1: Memory structure for Algorithm 8.4

8.5 Feasibility Problem With Computations

In the last section we gave the main ideas that allow reducing the storage memory. However, Algorithm 8.4 does not account for memory constraints in computations as per Definition 8.2. For instance, computing the volumetric center $\mathsf{VolumetricCenter}(\mathcal{P})$ already requires infinite memory for infinite precision. More importantly, even if one discretizes the queries, the necessary precision and computational power may be prohibitive with the classical Vaidya's method Algorithm 8.1. Even finding a feasible point in the polyhedron (let alone the volumetric center) using only the constraints is itself computationally intensive. There has been significant work to make Vaidya's method computationally tractable [Vai96; Ans97; Ans98]. These works address the issue of computational tractability, but the memory issue is still present. Indeed, the precision depends among other parameters on the condition number of the matrix $\boldsymbol{H}$ in order to compute the leverage scores σ_i for $i \in [m]$, which may not be well-conditioned. Second, to avoid memory overflow, we also need to ensure that the points queried have bounded norm, which is again not a priori guaranteed in the original version Algorithm 8.1.

To solve these issues and also give a computationally-efficient algorithm, the cutting-plane subroutine Algorithm 8.1 needs to be modified. In particular, the volumetric barrier needs to include regularization terms. Fortunately, these have already been studied in [LSW15]. In a major breakthrough, this work gave a cutting-plane algorithm with $\mathcal{O}(d^3 \ln^{\mathcal{O}(1)} \frac{d}{\epsilon})$ runtime complexity, improving over the seminal work from Vaidya and subsequent works which had $\mathcal{O}(d^{1+\omega} \ln^{\mathcal{O}(1)} \frac{d}{\epsilon})$ runtime complexity, where $\mathcal{O}(d^\omega)$ is the computational complexity of matrix multiplication. To achieve this result, they introduce various regularizing terms together with the logarithmic barrier. While the main motivation of [LSW15] was computational complexity, as a side effect, these regularization terms also ensure that computations can be carried with efficient memory. We then use their method as a subroutine.

For the sake of exposition and conciseness, we describe a simplified version of their method, that is also deterministic. This comes at the expense of a suboptimal running time

$\mathcal{O}(d^{1+\omega} \ln^{\mathcal{O}(1)} \frac{1}{\epsilon})$. We recall that our main concern is in memory usage rather than achieving the optimal runtime. The main technicality of this section is to show that their simplified method is numerically stable, and we emphasize that the original algorithm could also be shown to be numerically stable with similar techniques, leading to a time improvement from $\tilde{\mathcal{O}}(d^{1+\omega})$ to $\tilde{\mathcal{O}}(d^3)$. The memory usage, however, would not be improved.

8.5.1 A memory-efficient Vaidya's method for computations

Fix a polyhedron $\mathcal{P} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$. Using the same notations as for Vaidya's method in Section 8.4.1, we define the new leverage scores $\psi(\boldsymbol{x})_i = (\boldsymbol{A}_x(\boldsymbol{A}_x^\top \boldsymbol{A}_x + \lambda \boldsymbol{I})^{-1}\boldsymbol{A}_x^\top)_{i,i}$ and $\boldsymbol{\Psi}(\boldsymbol{x}) = diag(\boldsymbol{\psi}(\boldsymbol{x}))$. Let $\mu(\boldsymbol{x}) = \min_i \psi(\boldsymbol{x})_i$. Last, let $\boldsymbol{Q}(\boldsymbol{x}) = \boldsymbol{A}_x^\top(c_e \boldsymbol{I} + \boldsymbol{\Psi}(x))\boldsymbol{A}_x + \lambda \boldsymbol{I}$, where $c_e > 0$ is a constant parameter to be defined. In [LSW15], they consider minimizing the volumetric-analytic hybrid barrier function

$$p(\boldsymbol{x}) = -c_e \sum_{i=1}^{m} \ln s_i(\boldsymbol{x}) + \frac{1}{2} \ln \det(\boldsymbol{A}_x^\top \boldsymbol{A}_x + \lambda \boldsymbol{I}) + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2.$$

We can check [LSW15] that

$$\nabla p(\boldsymbol{x}) = -\boldsymbol{A}_x^\top(c_e \cdot \boldsymbol{1} + \boldsymbol{\psi}(x)) + \lambda \boldsymbol{x},$$

where $\boldsymbol{1}$ is the vector of ones. The following procedure gives a way to minimize this function efficiently given a good starting point.

Input: Initial point $\boldsymbol{x}^{(0)} \in \mathcal{P} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$
Input: Number of iterations $r > 0$
Given: $\|\nabla p(\boldsymbol{x}^{(0)})\|_{\boldsymbol{Q}(\boldsymbol{x}^{(0)})^{-1}} \leq \frac{1}{100}\sqrt{c_e + \mu(x^{(0)})} := \eta$.
1 **for** $k = 1$ *to* r **do**
2 **if** $\|\nabla p(\boldsymbol{x}^{(k-1)})\|_{\boldsymbol{Q}(\boldsymbol{x}^{(0)})^{-1}} \leq 2(1 - \frac{1}{64})^r \eta$ **then Break**;
3 $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} - \frac{1}{8}\boldsymbol{Q}(\boldsymbol{x}^{(0)})^{-1}\nabla p(\boldsymbol{x}^{(k-1)})$
4 **end**
Output: $\boldsymbol{x}^{(k)}$

Algorithm 8.5: $\boldsymbol{x}^{(r)} = \mathsf{Centering}(\boldsymbol{x}^{(0)}, r)$

We then present their simplified cutting-plane method.

In both Algorithm 8.5 and Algorithm 8.6, notice that the updates require to compute in particular the leverage scores $\boldsymbol{\psi}(\boldsymbol{x})$, which can be computed in $\mathcal{O}(d^\omega)$ time using their formula. To achieve the $\mathcal{O}(d^3 \ln^{\mathcal{O}(1)} \frac{1}{\epsilon})$ computational complexity, an amortized computational cost $\mathcal{O}(d^2)$ is needed. The algorithm from [LSW15] achieves this through various careful techniques aiming to update estimates of these leverage scores. The above cutting-plane algorithm is exactly that of [LSW15] when these estimates are always exact (i.e. recomputed at each iteration), which yields the $d^{\omega-2}$ overhead time complexity. In particular, the original proof of convergence and correctness of [LSW15] directly applies to this simplified algorithm.

It remains to check whether one can implement this algorithm with efficient memory, corresponding to checking this method's numerical stability.

Input: $\epsilon, \delta > 0$ and a separation oracle $O : \mathcal{C}_d \to \mathbb{R}^d$
Check: Throughout the algorithm, if $s_i(\boldsymbol{x}^{(t)}) < 2\epsilon$ for some i then **return** $(\mathcal{P}_t, \boldsymbol{x}^{(t)})$
Initialize $\boldsymbol{x}^{(0)} = \mathbf{0}$ and $\mathcal{P}_0 := \{(-1, \boldsymbol{e}_i, -1), (-1, -\boldsymbol{e}_i, -1),\ i \in [d]\}$
for $t \geq 0$ **do**
  if $\min_{i\in[m]} \psi(\boldsymbol{x}^{(t)})_i \leq c_d$ **then**
    $\mathcal{P}_{t+1} = \mathcal{P}_t \setminus \{(k_j, \boldsymbol{a}_j, b_j)\}$ where $j \in \arg\min_{i\in[m]} \psi(\boldsymbol{x}^{(t)})_i$
  else
    if $\boldsymbol{x}^{(t)} \notin \mathcal{C}_d$ **then** $\boldsymbol{a} = -sign(x_i)\boldsymbol{e}_i$ where $i \in \arg\min_{j\in[d]} |x_j^{(t)}|$;
    else $\boldsymbol{a} = O(\boldsymbol{x}^{(t)})$;
    Let $b = \boldsymbol{a}^\top \boldsymbol{x}^{(t)} - c_a^{-1/2} \sqrt{\boldsymbol{a}^\top (\boldsymbol{A}^\top \boldsymbol{S}_{x^{(t)}}^{-2} \boldsymbol{A} + \lambda \boldsymbol{I})^{-1} a}$
    $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{(t, \boldsymbol{a}, b)\}$
  $\boldsymbol{x}^{(t+1)} = \mathsf{Centering}(\boldsymbol{x}^{(t)}, 200, c_\Delta)$
end

Algorithm 8.6: An efficient cutting-plane method, simplified from [LSW15]

Lemma 8.6. *Suppose that each iterate of the centering Algorithm 8.5, $\|\nabla p(\boldsymbol{x}^{(k-1)})\|_{\boldsymbol{Q}(\boldsymbol{x}^{(0)})^{-1}}$ is computed up to precision $(1 - \frac{1}{64})^r \eta$ (l.2), and $\boldsymbol{x}^{(k)}$ is computed up to an error $\boldsymbol{\zeta}^{(k)}$ with $\|\boldsymbol{\zeta}^{(k)}\|_{\boldsymbol{Q}(\boldsymbol{x}^{(0)})} \leq \frac{1}{2^{10} r}(1 - \frac{1}{64})^r \eta$ (l.3).*

Then, Algorithm 8.5 outputs $\boldsymbol{x}^{(k)}$ such that $\|\nabla p(\boldsymbol{x}^{(k)})\|_{\boldsymbol{Q}^{-1}(\boldsymbol{x}^{(k)})} \leq 3(1 - \frac{1}{64})^r \eta$ and all iterates computed during the procedure satisfy $\|\boldsymbol{S}_{x^{(0)}}^{-1}(\boldsymbol{s}(\boldsymbol{x}^{(t)}) - \boldsymbol{s}(\boldsymbol{x}^{(0)}))\|_2 \leq \frac{1}{10}$.

Proof As mentioned above, without computation errors, the result from [LSW15] would apply directly. Here, we simply adapt the proof to the case with computational errors to show that it still applies. Denote $\boldsymbol{Q} = \boldsymbol{Q}(\boldsymbol{x}^{(0)})$ for convenience. Let $\eta = \frac{1}{100}\sqrt{c_e + \mu(\boldsymbol{x}^{(0)})}$. We prove by induction that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}} \leq 9\eta$, $\|\nabla p(\boldsymbol{x}^{(t)})\|_{\boldsymbol{Q}^{-1}} \leq (1 - \frac{1}{64})^t \eta$ for all $t \leq r$. For a given iteration t, denote $\tilde{\boldsymbol{x}}^{(t+1)} = \boldsymbol{x}^{(k-1)} - \frac{1}{8}\boldsymbol{Q}^{-1}\nabla p(\boldsymbol{x}^{(k-1)})$ the result of the exact computation. The same arguments as in the original proof give $\|\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}} \leq 9\eta$, and

$$\|\nabla p(\tilde{\boldsymbol{x}}^{(t+1)})\|_{\boldsymbol{Q}^{-1}} \leq \left(1 - \frac{1}{32}\right) \|\nabla p(\boldsymbol{x}^{(t)})\|_{\boldsymbol{Q}^{-1}}.$$

Now because $\|\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t+1)}\|_{\boldsymbol{Q}} \leq \eta$, we have $\|\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}}, \|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}} \leq 10\eta$, so that [LSW15, Lemma 11] gives $\boldsymbol{\nabla}^2 p(\boldsymbol{y}(u)) \preceq 8\boldsymbol{Q}(\boldsymbol{y}(u)) \preceq 16\boldsymbol{Q}$, where $\boldsymbol{y}(u) = \boldsymbol{x}^{(t+1)} + u(\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t+1)}$ for $u \in [0, 1]$. Thus,

$$\begin{aligned}\|\nabla p(\tilde{\boldsymbol{x}}^{(t+1)}) - \nabla p(\boldsymbol{x}^{(t+1)})\|_{\boldsymbol{Q}^{-1}} &\leq \left\| \int_0^1 \boldsymbol{\nabla}^2 p(\boldsymbol{y}(u))(\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t+1)}) \right\|_{\boldsymbol{Q}^{-1}} \\ &\leq 16 \|\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t+1)}\|_{\boldsymbol{Q}}.\end{aligned}$$

Now by construction of the procedure, if the algorithm performed iteration $t + 1$, we have $\|\nabla p(\boldsymbol{x}^{(t)})\|_{\boldsymbol{Q}^{-1}} \geq (1 - \frac{1}{64})^r \eta$. Combining this with the fact that $\|\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t+1)}\|_{\boldsymbol{Q}} \leq \frac{1}{2^{10} r}(1 -$

$\frac{1}{64})^r\eta$, obtain

$$\begin{aligned}\|\nabla p(\boldsymbol{x}^{(t+1)})\|_{\boldsymbol{Q}^{-1}} &\leq \|\nabla p(\tilde{\boldsymbol{x}}^{(t+1)}) - \nabla p(\boldsymbol{x}^{(t+1)})\|_{\boldsymbol{Q}^{-1}} + \|\nabla p(\tilde{\boldsymbol{x}}^{(t+1)})\|_{\boldsymbol{Q}^{-1}} \\ &\leq \left(1 - \frac{1}{64}\right) \|\nabla p(\boldsymbol{x}^{(t)})\|_{\boldsymbol{Q}^{-1}}.\end{aligned}$$

We now write

$$\begin{aligned}\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}} &\leq \sum_{k=0}^{t} \|\tilde{\boldsymbol{x}}^{(k+1)} - \boldsymbol{x}^{(k+1)}\|_{\boldsymbol{Q}} + \frac{1}{8}\|\boldsymbol{Q}^{-1}\nabla p(\boldsymbol{x}^{(k)})\|_{\boldsymbol{Q}} \\ &\leq \eta + \frac{1}{8}\sum_{i=0}^{\infty}\left(1 - \frac{1}{64}\right)^i \eta \leq 9\eta.\end{aligned}$$

The induction is now complete. When the algorithm stops, either the r steps were performed, in which case the induction already shows that $\|\nabla p(\boldsymbol{x}^{(r)})\|_{\boldsymbol{Q}^{-1}} \leq (1 - \frac{1}{64})^r\eta$. Otherwise, if the algorithm terminates at iteration k, because $\|\nabla p(\boldsymbol{x}^{(k)})\|_{\boldsymbol{Q}^{-1}}$ was computed to precision $(1 - \frac{1}{64})^r\eta$, we have (see l.2 of Algorithm 8.5)

$$\|\nabla p(\boldsymbol{x}^{(k)})\|_{\boldsymbol{Q}^{-1}} \leq 2\left(1 - \frac{1}{64}\right)^r \eta + \left(1 - \frac{1}{64}\right)^r \eta = 3\left(1 - \frac{1}{64}\right)^r \eta.$$

The same argument as in the original proof shows that at each iteration t,

$$\|\boldsymbol{S}_{x^{(0)}}^{-1}(\boldsymbol{s}(\boldsymbol{x}^{(t)}) - \boldsymbol{s}(\boldsymbol{x}^{(0)}))\|_2 = \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{A}^\top \boldsymbol{S}_{x^{(0)}}^{-2}\boldsymbol{A}} \leq \frac{\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(0)}\|_{\boldsymbol{Q}}}{\sqrt{\mu(\boldsymbol{x}^{(0)}) + c_e}} \leq \frac{1}{10}.$$

This ends the proof of the lemma. ■

Because of rounding errors, Lemma 8.6 has an extra factor 3 compared to the original guarantee in [LSW15, Lemma 14]. To achieve the same guarantee, it suffices to perform $70 \geq \ln(3)/\ln(1/(1 - \frac{1}{64}))$ additional centering procedures at most. hence, instead of performing 200 centering procedures during the cutting plane method, we perform 270 (l.10 of Algorithm 8.6). We next turn to the numerical stability of the main Algorithm 8.6.

Lemma 8.7. *Suppose that throughout the algorithm, when checking the stopping criterion* $\min_{i\in[m]} s_i(\boldsymbol{x}) < 2\epsilon$, *the quantities* $s_i(\boldsymbol{x})$ *were computed with accuracy* ϵ. *Suppose that at each iteration of Algorithm 8.6, the leverage scores* $\boldsymbol{\psi}(\boldsymbol{x}^{(t)})$ *are computed up to multiplicative precision* $c_\Delta/4$ *(l.3), that when a constraint is added, the response of the oracle* $\boldsymbol{a}$ *(l.7) is stored perfectly but* b *(l.8) is computed up to precision* $\Omega(\frac{\epsilon}{\sqrt{n}})$. *Further suppose that the centering Algorithm 8.5 is run with numerical approximations according to the assumptions in Lemma 8.6. Then, all guarantees for the original algorithm in [LSW15] hold, up to a factor* 3 *for* ϵ.

Proof We start with the termination criterion. Given the requirement on the computational accuracy, we know that the final output $\boldsymbol{x}$ satisfies $\min_{i\in[m]} s_i(\boldsymbol{x}) \leq 3\epsilon$. Further, during the algorithm, if it does not stop, then one has $\min_{i\in[m]} s_i(\boldsymbol{x}) \geq \epsilon$, which is precisely the guarantee of the original algorithm in [LSW15].

We next turn to the computation of the leverage scores in l.4. In the original algorithm, only a c_Δ-estimate is computed. Precisely, one computes a vector $\boldsymbol{w}^{(t)}$ such that for all $i \in [d]$, $\psi(\boldsymbol{x}^{(t)})_i \leq w_i \leq (1+c_\Delta)\psi(\boldsymbol{x}^{(t)})_i$, then deletes a constraint when $\min_{i\in[m^{(t)}]} w_i^{(t)} \leq c_d$. In the adapted algorithm, let $\tilde{\psi}(\boldsymbol{x}^{(t)})_i$ denote the computed leverage scores for $i \in [d]$. By assumption, we have

$$(1 - c_\Delta/4)\psi(\boldsymbol{x}^{(t)})_i \leq \tilde{\psi}(\boldsymbol{x}^{(t)})_i \leq (1 + c_\Delta/4)\psi(\boldsymbol{x}^{(t)})_i.$$

Up to re-defining the constant c_d as $(1 - c_\Delta/4)c_d$, $\tilde{\boldsymbol{\psi}}(\boldsymbol{x}^{(t)})$ is precisely within the guarantee bounds of the algorithm. For the accuracy on the separation oracle response and the second-term value b, [LSW15] emphasizes that the algorithm always changes constraints by a δ amount where $\delta = \Omega(\frac{\epsilon}{\sqrt{d}})$ so that an inexact separation oracle with accuracy $\Omega(\frac{\epsilon}{\sqrt{d}})$ suffices. Therefore, storing an $\Omega(\frac{\epsilon}{\sqrt{d}})$ accuracy of the second term keeps the guarantees of the algorithm. Last, we checked in Lemma 8.6 that the centering procedure Algorithm 8.5 satisfies all the requirements needed in the original proof [LSW15]. ■

For our recursive method, we need an efficient cutting-plane method that also provides a proof (certificate) of convergence. This is also provided by [LSW15] that provide a proof that the feasible region has small width in one of the directions $\boldsymbol{a}_i$ of the returned polyhedron.

Input: $\epsilon > 0$ and a separation oracle $O : \mathcal{C}_d \to \mathbb{R}^d$

1 Run Algorithm 8.6 to obtain a polyhedron $\mathcal{P}$ and a feasible point $\boldsymbol{x}$

2 $\boldsymbol{x}^\star = \mathsf{Centering}(\boldsymbol{x}, 64 \ln \frac{2}{\epsilon}, c_\Delta)$

3 $\lambda_i = \frac{c_e + \psi_i(\boldsymbol{x}^\star)}{s_i(\boldsymbol{x}^\star)} \left(\sum_j \frac{c_e + \psi_j(\boldsymbol{x}^\star)}{s_j(\boldsymbol{x}^\star)} \right)^{-1}$ for all i

Output: $(\mathcal{P}, \boldsymbol{x}^\star, (\lambda_i)_i)$

Algorithm 8.7: Cutting-plane algorithm with certified optimality

Lemma 8.8. *[LSW15, Lemma 28] Let $(\mathcal{P}, \boldsymbol{x}, (\lambda_i)_i)$ be the output of Algorithm 8.7. Then, $\boldsymbol{x}$ is feasible, $\|\boldsymbol{x}\|_2 \leq 3\sqrt{d}$, $\lambda_j \geq 0$ for all j and $\sum_i \lambda_i = 1$. Further,*

$$\left\| \sum_i \lambda_i \boldsymbol{a}_i \right\|_2 = \mathcal{O}\left(\epsilon \sqrt{d} \ln \frac{d}{\epsilon} \right), \quad \text{and} \quad \sum_i \lambda_i (\boldsymbol{a}_i^\top \boldsymbol{x} - b_j) \leq \mathcal{O}\left(d\epsilon \ln \frac{d}{\epsilon} \right).$$

We are now ready to show that Algorithm 8.6 can be implemented with efficient memory and also provides a proof of the convergence of the algorithm.

Proposition 8.2. *Provided that the output of the oracle are vectors discretized to precision $\mathrm{poly}(\frac{\epsilon}{d})$ and have norm at most 1, Algorithm 8.7 can be implemented with $\mathcal{O}(d^2 \ln \frac{d}{\epsilon})$ bits of memory to output a certified optimal point according to Lemma 8.8. The algorithm performs $\mathcal{O}(d \ln \frac{d}{\epsilon})$ calls to the separation oracle and runs in $\mathcal{O}(d^{1+\omega} \ln^{\mathcal{O}(1)} \frac{d}{\epsilon})$ time.*

Proof We already checked the numerical stability of Algorithm 8.6 in Lemma 8.7. It remains to check the next steps of the algorithm. The centering procedure is stable again via Lemma 8.6. It also suffices to compute the coefficients λ_j up to accuracy $\mathcal{O}(\epsilon/(\sqrt{d}) \ln(d/\epsilon))$ to keep the guarantees desired since by construction all vectors $\boldsymbol{a}_i$ have norm at most one.

It now remains to show that the algorithm can be implemented with efficient memory. We recall that at any point during the algorithm, the polyhedron $\mathcal{P}$ has at most $\mathcal{O}(d)$ constraints [LSW15, Lemma 22]. Hence, since we assumed that each vector $\boldsymbol{a}_i$ composing a constraint is discretized to precision $\text{poly}(\frac{\epsilon}{d})$, we can store the polyhedron constraints with $\mathcal{O}(d^2 \ln \frac{d}{\epsilon})$ bits of memory. The second terms b are computed up to precision $\Omega(\epsilon/\sqrt{d})$ hence only use $\mathcal{O}(d \ln \frac{d}{\epsilon})$ bits of memory. The algorithm also keeps the current iterate $x^{(t)}$ in memory. These are all bounded throughout the memory $\|x^{(t)}\|_2 = \mathcal{O}(\sqrt{d})$ [LSW15, Lemma 23], hence only require $\mathcal{O}(d \ln \frac{d}{\epsilon})$ bits of memory for the desired accuracy.

Next, the distances to the constraints are bounded at any step of the algorithm: $s_i(\boldsymbol{x}^{(t)}) \leq \mathcal{O}(\sqrt{d})$ [LSW15, Lemma 24], hence computing $s_i(\boldsymbol{x}^{(t)})$ to the required accuracy is memory-efficient. Recall that from the termination criterion, except for the last point, any point $\boldsymbol{x}$ during the algorithm satisfies $s_i(\boldsymbol{x}) \geq \epsilon$ for all constraints $i \in [m]$. In particular, this bounds the eigenvalues of $\boldsymbol{Q}$ since $\lambda I \preceq \boldsymbol{Q}(x) \preceq (\lambda + m(c_e + 1)/\epsilon^2)\boldsymbol{I}$. Thus, the matrix is sufficiently well-conditioned to achieve the accuracy guarantees from Lemma 8.6 using $\mathcal{O}(d^2 \ln \frac{d}{\epsilon})$ memory during matrix inversions (and matrix multiplications). Similarly, for the computation of leverage scores, we use $\boldsymbol{\Psi}(x) = diag(\boldsymbol{A}_x(\boldsymbol{A}_x^\top \boldsymbol{A}_x + \lambda \boldsymbol{I})^{-1}\boldsymbol{A}_x^\top)$, where $\lambda \boldsymbol{I} \preceq \boldsymbol{A}_x^\top \boldsymbol{A}_x + \lambda \boldsymbol{I} \preceq (\lambda + m\epsilon^{-2})\boldsymbol{I}$. This same matrix inversion appears when computing the second term of an added constraint. Overall, all linear algebra operations are well conditioned and implementable with required accuracy with $\mathcal{O}(d^2 \ln \frac{d}{\epsilon})$ memory. Using fast matrix multiplication, all these operations can be performed in $\tilde{\mathcal{O}}(d^\omega)$ time per iteration of the cutting-plane algorithm since these methods are also known to be numerically stable [DDH07]. Thus, the total time complexity is $\mathcal{O}(d^{1+\omega} \ln^{O(1)} \frac{d}{\epsilon})$. The oracle-complexity still has optimal $\mathcal{O}(d \ln \frac{d}{\epsilon})$ oracle-complexity as in the original algorithm. ■

Up to changing ϵ to $c \cdot \epsilon/(d \ln \frac{d}{\epsilon})$, the described algorithm finds constraints given by $\boldsymbol{a}_i$ and b_i, $i \in [m]$ returned by the normalized separation oracle, coefficients λ_i, $i \in [m]$, and a feasible point $\boldsymbol{x}^\star$ such that for any vector in the unit cube, $\boldsymbol{z} \in \mathcal{C}_d$, one has

$$\min_{i \in [m]} \boldsymbol{a}_i^\top \boldsymbol{z} - b_i \leq \sum_{i \in [m]} \lambda_i (\boldsymbol{a}_i^\top \boldsymbol{z} - b_i) \leq \left(\sum_{i \in [m]} \lambda \boldsymbol{a}_i \right)^\top (\boldsymbol{x}^\star - \boldsymbol{z}) + \sum_{i \in [m]} \lambda_i (\boldsymbol{a}_i^\top \boldsymbol{x}^\star - b_i) \leq \epsilon.$$

This effectively replaces Lemma 8.1.

8.5.2 Merging Algorithm 8.7 within the recursive algorithm

Algorithms 8.2 to 8.4 from the recursive procedure need to be slightly adapted to the new format of the cutting-plane method's output. In particular, the oracles do not take as input polyhedrons (and eventually query their volumetric center as before), but directly take as input an point (which is an approximate volumetric center).

The same proof as for Algorithm 8.4 shows that Algorithm 8.10 run with the parameters in Theorem 8.3 also outputs a successful vector using the same oracle-complexity. We only need to analyze the memory usage in more detail.

Proof of Theorem 8.1 As mentioned above, we will check that Algorithm 8.10 with the same parameters $\delta = \frac{\epsilon}{4d}$ and $\xi = \frac{\sigma_{min}\epsilon}{32d^{5/2}}$ as in Theorem 8.3 satisfies the desired require-

Input: δ, ξ, $O_x : \mathcal{C}_n \to \mathbb{R}^m$ and $O_y : \mathcal{C}_n \to \mathbb{R}^n$

Run Algorithm 8.7 with parameter $c \cdot \delta/(d \ln \frac{d}{\delta}), \xi$ and O_y to obtain $(\mathcal{P}^\star, \boldsymbol{x}^\star, \boldsymbol{\lambda})$

Store $\boldsymbol{k}^\star = (k_i, i \in [m])$ where $m = |\mathcal{P}^\star|$, and $\boldsymbol{\lambda}^\star \leftarrow \mathsf{Discretize}(\boldsymbol{\lambda}^\star, \xi)$

Initialize $\mathcal{P}_0 := \{(-1, \boldsymbol{e}_i, -1), (-1 - \boldsymbol{e}_i, -1),\ i \in [d]\}$, $\boldsymbol{x}^{(0)} = \mathbf{0}$ and let $\boldsymbol{u} = \mathbf{0} \in \mathbb{R}^m$

for $t = 0, 1, \ldots, \max_i k_i$ **do**

  if $t = k_i^\star$ *for some* $i \in [m]$ **then**

    $\boldsymbol{g}_x = O_x(\boldsymbol{x}^{(t)})$

    $\boldsymbol{u} \leftarrow \mathsf{Discretize}_m(\boldsymbol{u} + \lambda_i^\star \boldsymbol{g}_x, \xi)$

  Update $\mathcal{P}_t$ to get $\mathcal{P}_{t+1}$, and $\boldsymbol{x}^{(t)}$ to get $\boldsymbol{x}^{(t+1)}$ as in Algorithm 8.6

end

return $\boldsymbol{u}$

Algorithm 8.8: $\mathsf{ApproxSeparationVector}_{\delta,\xi}(O_x, O_y)$

Input: δ, ξ, $1 \le j \le i \le p$, $\boldsymbol{x}^{(r)} \in \mathcal{C}_{k_r}$ for $r \in [i]$, $O_S : \mathcal{C}_d \to \mathbb{R}^d$

if $i = p$ **then**

  $(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_p) = O_S(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$

  return $\mathsf{Discretize}_{k_j}(\boldsymbol{g}_j, \xi)$

end

Define $O_x : \mathcal{C}_{k_{i+1}} \to \mathbb{R}^{k_j}$ as $\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i+1, j, \boldsymbol{x}^{(1)}, , \ldots, \boldsymbol{x}^{(i)}, \cdot)$

Define $O_y : \mathcal{C}_{k_{i+1}} \to \mathbb{R}^{k_{i+1}}$ as $\mathsf{ApproxOracle}_{\delta,\xi,\mathcal{O}_f}(i+1, i+1, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)}, \cdot)$

return $\mathsf{ApproxSeparationVector}_{\delta,\xi}(O_x, O_y)$

Algorithm 8.9: $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(i, j, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)})$

ments. We have already checked its correctness and oracle-complexity. Using the same arguments, the computational complexity is of the form $\mathcal{O}(\mathcal{O}(\text{ComplexityCuttingPlanes})^p)$ where ComplexityCuttingPlanes is the computational complexity of the cutting-plane method used, i.e., here of Algorithm 8.7. Hence, the computational complexity is $\mathcal{O}((C(d/p)^{1+\omega} \ln^{\mathcal{O}(1)} \frac{d}{\epsilon})^p)$ for some universal constant $C \ge 2$. We now turn to the memory. In addition to the memory of Algorithm 8.4, described in Table 8.1, we need

1. a placement for all $i \in [p]$ for the current iterate $\boldsymbol{x}^{(i)}$: $\mathcal{O}(k_i \ln \frac{1}{\xi})$ bits,
2. a placement for computations, that is shared for all layers (used to compute leverage scores, centering procedures, etc. By Proposition 8.2, since the vectors are always discretized to precision ξ, this requires $\mathcal{O}(\max_{i \in [p]} k_i^2 \ln \frac{d}{\epsilon})$ bits,
3. the placement Q to perform queries is the concatenation of all of the placements $(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(p)})$: no additional bits needed.
4. a placement N to store the precision needed for the oracle responses: $\mathcal{O}(\ln \frac{1}{\xi})$ bits
5. a placement R to receive the oracle responses: $\mathcal{O}(d \ln \frac{1}{\xi})$ bits.

The new memory structure is summarized in Table 8.2.

Input: δ, ξ, and $\mathcal{O}_S : \mathcal{C}_d \to \mathbb{R}^d$ a separation oracle
Check: Throughout the algorithm, if O_S returned Success to a query $\boldsymbol{x}$, **return** $\boldsymbol{x}$
1 Run Algorithm 8.6 with parameters δ and ξ and oracle $\mathsf{ApproxOracle}_{\delta,\xi,O_S}(1, 1, \cdot)$

Algorithm 8.10: Memory-constrained algorithm for convex optimization

With the same arguments as in the original proof of Theorem 8.3, this memory is sufficient to run the algorithm and perform computations, thanks to the computation placement. The total number of bits used throughout the algorithm remains the same, $\mathcal{O}(\frac{d^2}{p} \ln \frac{d}{\epsilon})$. This ends the proof of the theorem. ■

i	1	...	p	Oracle response	Precision
j	$j^{(1)}$		$j^{(p)}$	$R = (R_1, \dots, R_p)$	N
Iteration index	$t^{(1)}$		$t^{(p)}$		
Polyhedron	$\mathcal{P}^{(1)} = \begin{pmatrix} k_1, \boldsymbol{a}_1, b_1 \\ k_2, \boldsymbol{a}_2, b_2 \\ \dots \\ k_m, \boldsymbol{a}_m, b_m \end{pmatrix}$		$\mathcal{P}^{(p)}$	Computation memory	
Current iterate	$\boldsymbol{x}^{(1)}$		$\boldsymbol{x}^{(p)}$		
Computed dual variables	$(\boldsymbol{k}^\star, \boldsymbol{\lambda}^\star) = \begin{pmatrix} k_1^\star, \lambda_1^\star \\ k_2^\star, \lambda_2^\star \\ \dots \end{pmatrix}$		$(\boldsymbol{k}^{\star(p)}, \boldsymbol{\lambda}^{\star(p)})$		
Working separation vector	$\boldsymbol{u}^{(1)}$		$\boldsymbol{u}^{(p)}$		

Table 8.2: Memory structure for Algorithm 8.10

8.6 Improved Oracle-Complexity/Memory Lower-Bound Trade-offs

We recall the three oracle-complexity/memory lower-bound trade-offs known in the literature.

1. First, [Mar+22] showed that any (including randomized) algorithm for convex optimization uses $d^{1.25-\delta}$ memory or makes $\tilde{\Omega}(d^{1+4\delta/3})$ queries.

2. Then, in Chapter 7 we showed that any deterministic algorithm for convex optimization uses $d^{2-\delta}$ memory or makes $\tilde{\Omega}(d^{1+\delta/3})$ queries.

3. Last, we also showed in Chapter 7 that any deterministic algorithm for the feasibility problem uses $d^{2-\delta}$ memory or makes $\tilde{\Omega}(d^{1+\delta})$ queries.

Although these works mainly focused on the regime $\epsilon = 1/\mathrm{poly}(d)$ and as a result $\ln \frac{1}{\epsilon} = \mathcal{O}(\ln d)$, neither of these lower bounds have an explicit dependence in ϵ. This can lead to sub-optimal lower bounds whenever $\ln \frac{1}{\epsilon} \gg \ln d$. Furthermore, in the exponential regime $\epsilon \leq \frac{1}{2^{\mathcal{O}(d)}}$, these results do not effectively give useful lower bounds. Indeed, in this regime, one has $d^2 = \mathcal{O}(d \ln \frac{1}{\epsilon})$ and as a result, the lower bounds provided are weaker than the classical $\Omega(d \ln \frac{1}{\epsilon})$ lower bounds for oracle-complexity [NYD83] and memory [WS19]. In particular, in this exponential regime, these results fail to show that there is any trade-off between oracle-complexity and memory.

In this section, we aim to explicit the dependence in ϵ of these lower-bounds. We show with simple modifications and additional arguments that one can roughly multiply these oracle-complexity and memory lower bounds by a factor $\ln \frac{1}{\epsilon}$ each. We split the proofs in two. First we give arguments to improve the memory dependence by a factor $\ln \frac{1}{\epsilon}$, which is achieved by modifying the sampling of the rows of the matrix $\boldsymbol{A}$ defining a wall term common to the functions considered in the lower bound proofs from [Mar+22] or Chapter 7. Then we show how to improve the oracle-complexity dependence by an additional $\ln \frac{1}{\epsilon} / \ln d$ factor, via a standard rescaling argument.

8.6.1 Improving the memory lower bound

We start with some concentration results on random vectors. [Mar+22] gave the following result for random vectors in the hypercube.

Lemma 8.9 ([Mar+22])**.** *Let $\boldsymbol{h} \sim \mathcal{U}(\{\pm 1\}^d)$. Then, for any $t \in (0, 1/2]$ and any matrix $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_k] \in \mathbb{R}^{d \times k}$ with orthonormal columns,*

$$\mathbb{P}(\|\boldsymbol{Z}^\top \boldsymbol{h}\|_\infty \leq t) \leq 2^{-c_H k}.$$

Instead, we will need a similar concentration result for random unit vectors in the unit sphere.

Lemma 8.10. *Let $k \leq d$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ be k orthonormal vectors, and $\zeta \leq 1$.*

$$\mathbb{P}_{\boldsymbol{y} \sim \mathcal{U}(S^{d-1})} \left(|\boldsymbol{x}_i^\top \boldsymbol{y}| \leq \frac{\zeta}{\sqrt{d}},\ i \in [k] \right) \leq \left(\frac{2}{\sqrt{\pi}} \zeta \right)^k \leq (\sqrt{2}\zeta)^k.$$

Proof First, by isometry, we can suppose that the orthonormal vectors are simply $\boldsymbol{e}_1, \dots, \boldsymbol{e}_k$. We now prove the result by induction on d. For $d = 1$, the result holds directly. Fix $d \geq 2$, and $1 \leq k < d$. Then, if S_n is the surface area of S^n the n-dimensional sphere, then

$$\mathbb{P}\left(|y_1| \leq \frac{\zeta}{\sqrt{d}} \right) \leq \frac{S_{d-2}}{S_{d-1}} \frac{2\zeta}{\sqrt{d}} = \frac{2\zeta}{\sqrt{\pi d}} \frac{\Gamma(d/2)}{\Gamma(d/2 - 1/2)} \leq \frac{2}{\sqrt{\pi}} \zeta. \tag{8.7}$$

Conditionally on the value of y_1, the vector $(y_2, \dots, y_d)$ follows a uniform distribution on the $(d-2)$-sphere of radius $\sqrt{1 - y_1^2}$. Then,

$$\mathbb{P}\left(|y_i| \leq \frac{\zeta}{\sqrt{d}},\ 2 \leq i \leq k \mid y_1 \right) = \mathbb{P}_{\boldsymbol{z} \sim \mathcal{U}(S^{d-2})} \left(|z_i| \leq \frac{\zeta}{\sqrt{d(1 - y_1^2)}},\ 2 \leq i \leq k \right)$$

Now recall that since $|x_1| \leq 1/\sqrt{d}$, we have $d(1-x_1^2) \geq d-1$. Therefore, using the induction,

$$\mathbb{P}\left(|y_i| \leq \frac{\zeta}{\sqrt{d}},\, 2 \leq i \leq k \mid y_1\right) \leq \mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}(S^{d-2})}\left(|z_i| \leq \frac{\zeta}{\sqrt{d-1}},\, 2 \leq i \leq k\right) \leq \left(\frac{2\zeta}{\sqrt{\pi}}\right)^{k-1}.$$

Combining this equation with Eq (8.7) ends the proof. ■

We next use the following lemma to partition the unit sphere S^{d-1}.

Lemma 8.11 ([FS02] Lemma 21)**.** *For any $0 < \delta < \pi/2$, the sphere S^{d-1} can be partitioned into $N(\delta) = (\mathcal{O}(1)/\delta)^d$ equal volume cells, each of diameter at most δ.*

Following the notation from Chapter 7, we denote by $\mathcal{V}_\delta = \{V_i(\delta), i \in [N(\delta)]\}$ the corresponding partition, and consider a set of representatives $\mathcal{D}_\delta = \{\boldsymbol{b}_i(\delta), i \in [N(\delta)]\} \subset S^{d-1}$ such that for all $i \in [N(\delta)]$, $\boldsymbol{b}_i(\delta) \in V_i(\delta)$. With these notations we can define the discretization function ϕ_δ as follows

$$\phi_\delta(\boldsymbol{x}) = \boldsymbol{b}_i(\delta), \quad \boldsymbol{x} \in V_i(\delta).$$

We then denote by $\mathcal{U}_\delta$ the distribution of $\phi_\delta(\boldsymbol{z})$ where $\boldsymbol{z} \sim \mathcal{U}(S^{d-1})$ is sampled uniformly on the sphere. Note that because the cells of $\mathcal{V}_\delta$ have equal volume, $\mathcal{U}_\delta$ is simply the uniform distribution on the discretization $\mathcal{D}_\delta$.

We are now ready to give the modifications necessary to the proofs, to include a factor $\ln \frac{1}{\epsilon}$ for the necessary memory. For their lower bounds, [Mar+22] exhibit a distribution of convex functions that are hard to optimize. Building upon their work in Chapter 7 we constructed classes of convex functions that are hard to optimize, but that also depend adaptively on the considered optimization algorithm. For both, the functions considered a barrier term of the form $\|\boldsymbol{A}\boldsymbol{x}\|_\infty$, where $\boldsymbol{A}$ is a matrix of $\approx d/2$ rows that are independently drawn as uniform on the hypercube $\mathcal{U}(\{\pm 1\}^d)$. The argument shows that memorizing $\boldsymbol{A}$ is necessary to a certain extent. As a result, the lower bounds can only apply for a memory of at most $\mathcal{O}(d^2)$ bits, which is sufficient to memorize such a binary matrix. Instead, we draw rows independently according to the distribution $\mathcal{U}_\delta$, where $\delta \approx \epsilon$. We explicit the corresponding adaptations for each known trade-off. We start with the lower bounds from Chapter 7 for ease of exposition; although these build upon those of [Mar+22], their parametrization makes the adaptation more straightforward.

Lower bound of Chapter 7 for convex optimization and deterministic algorithms

For this lower bound, we use the exact same form of functions as they introduced,

$$\max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \eta\boldsymbol{v}_0^\top\boldsymbol{x}, \eta\left(\max_{p\leq p_{max}, l\leq l_p} \boldsymbol{v}_{p,l}^\top\boldsymbol{x} - p\gamma_1 - l\gamma_2\right)\right\},$$

with the difference that rows of $\boldsymbol{A}$ are take i.i.d. distributed according to $\mathcal{U}_{\delta'}$ instead of $\mathcal{U}(\{\pm 1\}^d)$. As a remark, they use $n = \lceil d/4 \rceil$ rows for $\boldsymbol{A}$. Except for η, we keep all parameters γ_1, γ_2, etc as in the original proof, and we will take $\delta' = \epsilon$ and $\eta = 2\sqrt{d}\epsilon$. The reason why we introduced δ' instead of δ is that the original construction also needs the discretization ϕ_δ. This is used during the optimization procedure which constructs adaptively this class of functions, and only needs $\delta = \text{poly}(1/d)$ instead of δ of order ϵ.

Theorem 8.4. *For $\epsilon \leq 1/(2d^{4.5})$ and any $\delta \in [0,1]$, a deterministic first-order algorithm guaranteed to minimize 1-Lipschitz convex functions over the unit ball with ϵ accuracy uses at least $d^{2-\delta} \ln \frac{1}{\epsilon}$ bits of memory or makes $\tilde{\Omega}(d^{1+\delta/3})$ queries.*

With the changes defined above, we can easily check that all results from Chapter 7 which reduce convex optimization to the optimization procedure, then the optimization procedure to the Orthogonal Vector Game with Hints (OVGH), Game 7.2 from Chapter 7, are not affected by our changes. The only modifications to perform are to the proof of query lower bound for the OVGH, Proposition 7.4 from Chapter 7. We emphasize that the distribution of $\boldsymbol{A}$ is changed in the optimization procedure but also in OVGH as a result.

Proposition 8.3. *Let $k \geq 20 \frac{M+3d\log(2d)+1}{n \log_2(\sqrt{2}(\zeta+\delta'\sqrt{d}))^{-1}}$. Let $0 < \alpha, \beta \leq 1$ such that $\alpha(\sqrt{d}/\beta)^{5/4} \leq \zeta/\sqrt{d}$ where $\zeta \leq 1$. If the Player wins the adapted OVGH with probability at least $1/2$, then $m \geq \frac{1}{8}(1 + \frac{30\log_2 d}{\log_2(\sqrt{2}(\zeta+\delta'\sqrt{d}))^{-1}})^{-1} d$.*

Proof We use the same proof and only highlight the modifications. The proof is unchanged until the step when the concentration result Lemma 8.9 is used. Instead, we use Lemma 8.10. With the same notations as in the original proof, we constructed $\lceil k/5 \rceil$ orthonormal vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_{\lceil k/5 \rceil}]$ such that all rows $\boldsymbol{a}$ of $\boldsymbol{A}'$ (which is $\boldsymbol{A}$ up to some observed and unimportant rows) one has

$$\|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \frac{\zeta}{\sqrt{d}}.$$

Next, by Lemma 8.10, we have

$$\begin{aligned}
\left| \left\{ \boldsymbol{a} \in \mathcal{D}_{\delta'} : \|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \frac{\zeta}{\sqrt{d}} \right\} \right| &\leq |\mathcal{D}_{\delta'}| \cdot \mathbb{P}_{\boldsymbol{a} \sim \mathcal{U}_{\delta'}} \left(\|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \frac{\zeta}{\sqrt{d}} \right) \\
&\leq |\mathcal{D}_{\delta'}| \cdot \mathbb{P}_{\boldsymbol{z} \sim \mathcal{U}(S^{d-1})} \left(\|\boldsymbol{Z}^\top \boldsymbol{z}\|_\infty \leq \frac{\zeta}{\sqrt{d}} + \delta' \right) \\
&\leq |\mathcal{D}_{\delta'}| \cdot \left(\sqrt{2}(\zeta + \delta'\sqrt{d}) \right)^{\lceil k/5 \rceil}.
\end{aligned}$$

Hence, using the same arguments as in the original proof, we obtain

$$H(\boldsymbol{A}' \mid \boldsymbol{Y}) \leq (n-m) \left(\log_2 |\mathcal{D}_{\delta'}| + \mathbb{P}(\mathcal{E}) \cdot \frac{k}{5} \log_2 \left(\sqrt{2}(\zeta + \delta'\sqrt{d}) \right) \right),$$

where $\mathcal{E}$ is the event when the algorithm succeeds at the OVGH game. In the next step, we need to bound $H(\boldsymbol{A} \mid \boldsymbol{V}) - H(\boldsymbol{G}, \boldsymbol{j}, \boldsymbol{c})$ where $\boldsymbol{V}$ stores hints received throughout the game, $\boldsymbol{G}$ stores observed rows of $\boldsymbol{A}$ during the game, and $\boldsymbol{j}, \boldsymbol{c}$ are auxiliary variables. The latter can be treated as in the original proof. We obtain

$$\begin{aligned}
H(\boldsymbol{A} \mid \boldsymbol{V}) - H(\boldsymbol{G}, \boldsymbol{j}, \boldsymbol{c}) &\geq H(\boldsymbol{A}) - H(\boldsymbol{G}) - I(\boldsymbol{A}; \boldsymbol{V}) - 3m \log_2(2d) \\
&\geq (n-m) \log_2 |\mathcal{D}_{\delta'}| - 3m \log_2(2d) - I(\boldsymbol{A}, \boldsymbol{V}).
\end{aligned}$$

Now the same arguments as in the original proof show that we still have $I(\boldsymbol{A}, \boldsymbol{V}) \leq 3km \log_2 d + 1$, and that as a result, if M is the number of bits stored in memory,

$$M \geq \frac{k}{10} \log_2 \left(\frac{1}{\sqrt{2}(\zeta + \delta'\sqrt{d})} \right) (n-m) - 3km \log_2 d - 1 - 3d \log_2(2d).$$

Then, with the same arguments as in the original proof, we can conclude. ■

We are now ready to prove Theorem 8.4. With the parameter

$$k = \lceil 20 \frac{M + 3d\log(2d) + 1}{n \log_2(\sqrt{2}(\epsilon d^4/2 + \delta'\sqrt{d}))^{-1}} \rceil$$

and the same arguments, we show that an algorithm solving the convex optimization up to precision $\eta/(2\sqrt{d}) = \epsilon$ yields an algorithm solving the OVGH where the parameters $\alpha = \frac{2\eta}{\gamma_1}$ and $\beta = \frac{\gamma_2}{4}$ satisfy

$$\alpha \left(\frac{\sqrt{d}}{\beta}\right)^{5/4} \leq \frac{\eta d^3}{4} = \frac{d^{3.5}\epsilon}{2}.$$

We can then apply Proposition 8.3 with $\zeta = d^4\epsilon/2$. Hence, if Q is the maximum number of queries of the convex optimization algorithm, we obtain

$$\lceil Q/p_{max} \rceil + 1 \geq \frac{1}{8}\left(1 + \frac{30\log_2 d}{\log_2 \frac{1}{d^4\epsilon} - 1/2}\right)^{-1} d \geq \frac{d}{8 \cdot 61},$$

where in the last inequality we used $\epsilon \leq 1/(2d^{4.5})$. As a result, with the same arguments, we obtain

$$Q = \Omega\left(\frac{d^{5/3}\ln^{1/3}\frac{1}{\epsilon}}{(M + \ln d)^{1/3}\ln^{2/3} d}\right).$$

This ends the proof of Theorem 8.4.

Lower bound ofChapter 7 for feasibility problems and deterministic algorithms

We improve the memory dependence by showing the following result.

Theorem 8.5. *For $\epsilon = 1/(48d^3)$ and any $\delta \in [0, 1]$, a deterministic algorithm guaranteed to solve the feasibility problem over the unit ball with ϵ accuracy uses at least $d^{2-\delta}\ln\frac{1}{\epsilon}$ bits of memory or makes at least $\tilde{\Omega}(d^{1+\delta})$ queries.*

We use the exact same class of feasibility problems and only change the parameter η_0 which constrained successful points to satisfy $\|\boldsymbol{A}\boldsymbol{x}\|_\infty \leq \eta_0$, as well as the rows of $\boldsymbol{A}$ that are sampled i.i.d. from $\mathcal{U}_\delta$. The other parameter $\eta_1 = 1/(2\sqrt{d})$ is unchanged. We also take $\delta' = \epsilon$. Because the rows of $\boldsymbol{A}$ are already normalized, we can take $\eta_0 = \epsilon$ directly. Then, the same proof as in Chapter 7 shows that if an algorithm solves feasibility problems with accuracy ϵ, there is an algorithm for OVGH for parameters $\alpha = \eta/\eta_1$ and $\beta = \eta_1/2$. Then, we have $\alpha(\sqrt{d}/\beta)^{5/4} \leq 12d^2\eta_0$ and we can apply Proposition 8.3 with $\zeta = 12d^{2.5}\eta_0 = 12d^{2.5}\epsilon$. Similar computations as above then show that $m \geq d/(8 \cdot 61)$, with $k = \Theta(\frac{M+\ln d}{d\ln\frac{1}{\epsilon}})$, so that the query lower bound finally becomes

$$Q \geq \Omega\left(\frac{d^3\ln\frac{1}{\epsilon}}{(M + \ln d)\ln^2 d}\right).$$

Remark 8.2. *The more careful analysis—involving the discretization $\mathcal{D}_\delta$ of the unit sphere at scale δ instead of the hypercube $\{\pm 1\}^d$—allowed to add a $\ln \frac{1}{\epsilon}$ factor to the final query lower bound but also an additional $\ln d$ factor for both convex-optimization and feasibility-problem results. Indeed, the improved Proposition 8.3 shows that the OVGH with adequate parameters requires $\mathcal{O}(d)$ queries, instead of $\mathcal{O}(d/\ln d)$ in Proposition 7.4 from Chapter 7. At a high level, each hint queried brings information $\mathcal{O}(d \ln d)$ but memorizing a binary matrix $\boldsymbol{A} \in \{\pm 1\}^{\lceil d/4 \rceil \times d}$ only requires d^2 bits of memory: hence the query lower bound is limited to $\mathcal{O}(d/\ln d)$. Instead, memorizing the matrix $\boldsymbol{A}$ where each row lies in $\mathcal{D}_\delta$ requires $\Theta(d^2 \ln \frac{1}{\epsilon})$ memory, hence querying d hints (total information $\mathcal{O}(d^2 \ln d)$) is not prohibitive for the lower bound.*

Lower bound of [Mar+22] for convex optimization and randomized algorithms

We aim to improve the result to obtain the following.

Theorem 8.6. *For $\epsilon \leq 1/d^4$ and any $\delta \in [0, 1]$, any (potentially randomized) algorithm guaranteed to minimize 1-Lipschitz convex functions over the unit ball with ϵ accuracy uses at least $d^{1.25-\delta} \ln \frac{1}{\epsilon}$ bits of memory or makes $\tilde{\Omega}(d^{1+4\delta/3})$ queries.*

The distribution considered in [Mar+22] is given by the functions

$$\frac{1}{d^6} \max \left\{ d^5 \|\boldsymbol{A}\boldsymbol{x}\|_\infty - 1, \max_{i \in [N]} (\boldsymbol{v}_i^\top \boldsymbol{x} - i\gamma) \right\},$$

where $N \leq d$ is a parameter, $\boldsymbol{A}$ has $\lfloor d/2 \rfloor$ rows drawn i.i.d. from $\mathcal{U}(\{\pm 1\}^d)$, and the vectors $\boldsymbol{v}_i$ are drawn i.i.d. from the rescaled hypercube $\boldsymbol{v}_i \sim \mathcal{U}(d^{-1/2}\{\pm 1\}^d)$. We adapt the class of functions by simply changing pre-factors as follows

$$\mu \max \left\{ \frac{1}{\mu} \|\boldsymbol{A}\boldsymbol{x}\|_\infty - 1, \max_{i \in [N]} (\boldsymbol{v}_i^\top \boldsymbol{x} - i\gamma) \right\}, \tag{8.8}$$

where $\boldsymbol{A}$ has the same number of rows but they are draw i.i.d. from $\mathcal{U}_\delta$, and $\delta, \mu > 0$ are parameters to specify. We use the notation μ instead of η as in the previous sections because [Mar+22] already use a parameter η which in our context can be interpreted as $\eta = 1/(\mu\sqrt{d})$. We choose the parameters $\mu = 16\sqrt{d}\epsilon$ and $\delta' = \epsilon$.

Again, as for the previous sections, the original proof can be directly used to show that if an algorithm is guaranteed to find a $\frac{\mu}{16\sqrt{N}}(\geq \epsilon)$-suboptimal point for the above function class, there is an algorithm that wins at their Orthogonal Vector Game (OVG) [Mar+22, Game 1], with the only difference that the parameter d^{-4} (l.8 of OVG) is replaced by $\sqrt{d}\mu$. OVG requires the output to be *robustly-independent* (defined in [Mar+22]) and effectively corresponds to $\beta = 1/d^2$ in OVGH. As a result, there is a successful algorithm for the OVGH with parameters $\alpha = \sqrt{d}\mu$ and $\beta = 1/d^2$ and that even completely ignores the hints. Hence, we can now directly use Proposition 8.3 with $\zeta = d^{1+25/16}\mu$ (from the assumption $\epsilon \leq d^{-4}$ we have $\zeta \leq 1/\sqrt{d}$). This shows that with the adequate choice of $k = \Theta(\frac{M + d \ln d}{d \ln \frac{1}{\epsilon}})$, the query lower bound is $\Omega(d)$.

Putting things together, a potentially randomized algorithm for convex optimization that uses M memory makes at least the following number of queries

$$Q \geq \Omega\left(\frac{Nd}{k}\right) = \Omega\left(\frac{d^{4/3}}{\ln^{1/3} d}\left(\frac{d \ln \frac{1}{\epsilon}}{M + d \ln d}\right)^{4/3}\right).$$

8.6.2 Proof sketch for improving the query-complexity lower bound

We now turn to improving the query-complexity lower bound by a factor $\frac{\ln \frac{1}{\epsilon}}{\ln d}$. At the high level, the idea is to replicate these constructed "difficult" class of functions at $\frac{\ln \frac{1}{\epsilon}}{\ln d}$ different scales or levels, similarly to the manner that the historical $\Omega(d \ln \frac{1}{\epsilon})$ lower bound is obtained for convex optimization [NYD83]. This argument is relatively standard and we only give details in the context of improving the bound from [Mar+22] for randomized algorithms in convex optimization for conciseness. This result uses a simpler class of functions, which greatly eases the exposition. We first present the construction with 2 levels, then present the generalization to $p = \Theta(\frac{\ln \frac{1}{\epsilon}}{\ln d})$ levels. For convenience, we write

$$Q(\epsilon; M, d) = \Omega\left(\frac{d^{4/3}}{\ln^{1/3} d}\left(\frac{d \ln \frac{1}{\epsilon}}{M + d \ln d}\right)^{4/3}\right).$$

This is the query lower bound given in Theorem 8.7 for convex optimization algorithms with memory M that optimize the defined class of functions (Eq (8.8)) to accuracy ϵ.

Construction of a bi-level class of functions $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ to optimize

In the lower-bound proof, [Mar+22] introduce the point

$$\bar{\boldsymbol{x}} = -\frac{1}{2\sqrt{N}} \sum_{i \in [N]} P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_i),$$

where $P_{\boldsymbol{A}^\perp}$ is the projection onto the orthogonal space to the rows of $\boldsymbol{A}$. They show that with failure probability at most $2/d$, $\bar{\boldsymbol{x}}$ has good function value

$$F_{\boldsymbol{A},\boldsymbol{v}}(\bar{\boldsymbol{x}}) := \mu \max\left\{\frac{1}{\mu}\|\boldsymbol{A}\bar{\boldsymbol{x}}\|_\infty - 1, \max_{i \in [N]}(\boldsymbol{v}_i^\top \bar{\boldsymbol{x}} - i\gamma)\right\} \leq -\frac{\mu}{8\sqrt{N}}.$$

This is shown in [Mar+22, Lemma 25]. On the other hand, from Theorem 8.6, during the first

$$Q_1 = Q(\epsilon; M, d)$$

queries of any algorithm, with probability at least $1/3$, all queries are at least $\mu/(16\sqrt{N})$-suboptimal compared to $\bar{\boldsymbol{x}}$ in function value [Mar+22, Theorem 28, Lemma 14 and Theorem 16]. Precisely, if $F_{\boldsymbol{A},\boldsymbol{v}}$ is the sampled function to optimize, with probability at least $1/3$,

$$F_{\boldsymbol{A},\boldsymbol{v}}(\boldsymbol{x}_t) \geq F_{\boldsymbol{A},\boldsymbol{v}}(\bar{\boldsymbol{x}}) + \frac{\mu}{16\sqrt{N}} \geq F_{\boldsymbol{A},\boldsymbol{v}}(\bar{\boldsymbol{x}}) + \frac{\mu}{16\sqrt{d}}, \quad \forall t \leq Q_1.$$

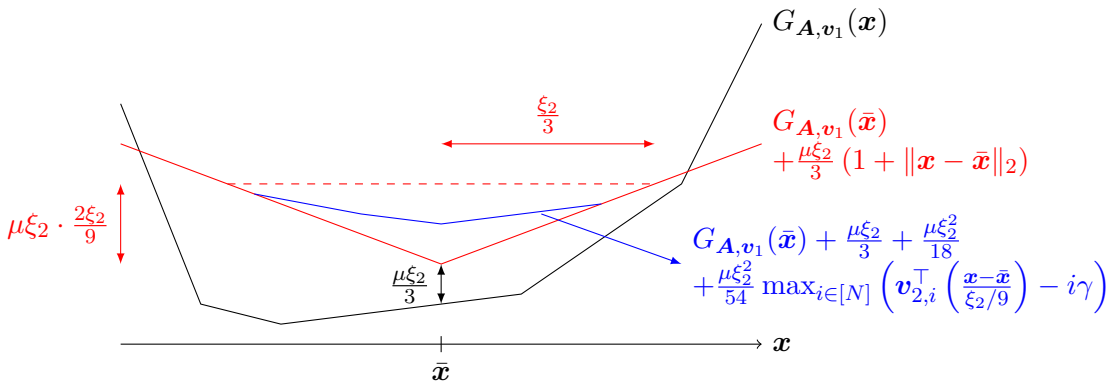

Figure 8.4: Representation of the procedure to rescale the optimization function.

As a result, we can replicate the term $\max_{i\in[N]}(\boldsymbol{v}_i^\top\boldsymbol{x}-i\gamma)$ at a smaller scale within the ball $B_d(\bar{\boldsymbol{x}},1/(16\sqrt{d}))$. For convenience, we introduce $\xi_2 = 1/(16\sqrt{d})$ which will be the scale of the duplicate function. We separate the wall term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \mu$ for convenience. Hence, we define

$$
\begin{aligned}
G_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x}) &:= \mu \max_{i\in[N]}\left(\boldsymbol{v}_{1,i}^\top\boldsymbol{x}-i\gamma\right)\\
G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x}) &:= \max\{G_{\boldsymbol{A},\boldsymbol{v}^{(1)}}(\boldsymbol{x}), G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + \frac{\mu\xi_2}{3}\cdot\\
&\qquad \max\left\{1+\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2, 1+\frac{\xi_2}{6}+\frac{\xi_2}{18}\max_{i\in[N]}\left(\boldsymbol{v}_{2,i}^\top\left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}}{\xi_2/9}\right)-i\gamma\right)\right\}\right\}
\end{aligned}
$$

An illustration of the construction is given in Fig. 8.4. The resulting optimization functions are given by adding the wall term:

$$
\begin{aligned}
F_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x}) &= \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty-\mu, G_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x})\right\}\\
F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x}) &= \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty-\mu, G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x})\right\}
\end{aligned}
$$

We first explain the choice of parameters. First observe that since $\|\boldsymbol{A}\bar{\boldsymbol{x}}\| = 0$, we have $G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) = F_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}})$. We can then check that for all $\boldsymbol{x}\in B_d(0,1)$,

$$
G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x}) \le \max\left\{G_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x}), G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}})+\frac{2}{3}\mu\xi_2\right\}. \tag{8.9}
$$

Further, for any $\boldsymbol{x}\in B_d(\bar{\boldsymbol{x}},\xi_2/3)$, since $F_{\boldsymbol{A},\boldsymbol{v}_1}$ is 1-Lipschitz, we can easily check that

$$
\begin{aligned}
&G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x}) - G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}})\\
&\qquad = \frac{\mu\xi_2}{3}\max\left\{1+\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2, 1+\frac{\xi_2}{6}+\frac{\xi_2}{18}\max_{i\in[N]}\left(\boldsymbol{v}_{2,i}^\top\left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}}{\xi_2/9}\right)-i\gamma\right)\right\} \le \frac{2}{3}\mu\xi_2.
\end{aligned}
$$

Thus, $G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x})$ does not coincide with $G_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x})$ on $B_d(\bar{\boldsymbol{x}},\xi_2/3)$. Then, the $\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2$ term ensures that any minimizer of $G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ is contained within the closed ball $B_d(\bar{\boldsymbol{x}},\xi_2/3)$. Also, to obtain a $\mu\xi_2/3$-suboptimal solution of $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$, the algorithm needs to find what would

be a $\mu\xi_2$-suboptimal solution of $F_{\boldsymbol{A},\boldsymbol{v}_1}$, while receiving the same response as when optimizing the latter. Next, for any $\boldsymbol{x} \in B_d(\bar{\boldsymbol{x}}, \xi_2/9)$, the term $\max_{i\in[N]} \left(\boldsymbol{v}_{2,i}^\top \left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}}{\xi_2/9}\right) - i\gamma\right)$ lies in $[-1,1]$. Hence, we can check that for $\boldsymbol{x} \in B_d(\bar{\boldsymbol{x}}, \xi_2/9)$,

$$G_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}(\boldsymbol{x}) = G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + \frac{\mu\xi_2}{3} + \frac{\mu\xi_2^2}{18} + \frac{\mu\xi_2^2}{54} \max_{i\in[N]} \left(\boldsymbol{v}_{2,i}^\top \left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}}{\xi_2/9}\right) - i\gamma\right). \tag{8.10}$$

We now argue that $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ acts as a duplicate function. Until the algorithm reaches a point with function value at most $G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + \mu\xi_2$, the optimization algorithm only receives responses consistent with the function $F_{\boldsymbol{A},\boldsymbol{v}_1}$ by Eq (8.9). Next, all minimizers of $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ are contained in $B_d(\bar{\boldsymbol{x}}, \xi_2/3)$, which was the goal of introducing the term in $\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2$. As a result, optimizing $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ on this ball is equivalent to minimizing

$$\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}(\boldsymbol{y}) = \max\left\{\|\boldsymbol{A}\boldsymbol{y}\|_\infty - \mu_2, c_2 + \nu_2 \max_{i\in[N]}(\boldsymbol{v}_{2,i}^\top \boldsymbol{y} - i\gamma), c_2' + \nu_2'\|\boldsymbol{y}\|\right\}, \quad \boldsymbol{y} \in B_d(0,3),$$

where $\boldsymbol{y} = \frac{\boldsymbol{x}-\bar{\boldsymbol{x}}}{\xi_2/9}$. The function has been rescaled by a factor $\xi_2/9$ compared to $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ so that $\mu_2 = \frac{9\mu}{\xi_2}$, $\nu_2 = \frac{\mu\xi_2}{6}$, $\nu_2' = 6\mu$, $c_2 = \frac{9}{\xi_2}G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + 3\mu + \frac{\mu\xi_2}{2}$, and $c_2' = \frac{9}{\xi_2}G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + 3\mu$. By Eq (8.10), the two first terms of $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_1}$ are preponderant for $\boldsymbol{y} \in B_d(0,1)$.

The form of $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$ is very similar to the original form of functions

$$F_{\boldsymbol{A},\boldsymbol{v}_2} = \max\left\{\|\boldsymbol{A}\boldsymbol{y}\|_\infty - \mu_1', \mu_2' \max_{i\in[N]}(\boldsymbol{v}_{2,i}^\top \boldsymbol{y} - i\gamma)\right\},$$

In fact, the same proof structure for the query-complexity/memory lower-bound can be applied in this case. The main difference is that originally one had $\mu_1' = \mu_2'$; here we would instead have $\mu_1' = \mu_2 + c_2 = \Theta(\mu/\xi_2)$ and $\mu_2' = \nu_2 = \Theta(\mu\xi_2)$. Intuitively, this corresponds to increasing the accuracy to $\Theta(\epsilon\xi_2^2)$—a factor ξ_2 is due to the fact that $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$ was rescaled by a factor $\xi_2/9$ compared to $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$, and a second factor ξ_2 is due to the fact that within $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$, we have $\mu_2' = \Theta(\mu\xi_2)$—while the query lower bound is similar to that obtained for $\Theta(\epsilon/\xi_2)$. As a result, during the first

$$Q_2 = Q\left(\Theta\left(\frac{\epsilon}{\xi_2}\right); M, d\right)$$

queries of any algorithm optimizing $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$, with probability at least $1/3$ on the sample of $\boldsymbol{A}$ and $\boldsymbol{v}_2$, all queries are at least $\Theta(\epsilon\xi_2)$-suboptimal compared to

$$\bar{\boldsymbol{y}} = -\frac{1}{2\sqrt{N}} \sum_{i\in[N]} P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{2,i}).$$

We are now ready to give lower bounds on the queries of an algorithm minimizing $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ to accuracy $\Theta(\epsilon\xi_2^2)$. Let T_2 be the index of the first query with function value at most $G_{\boldsymbol{A},\boldsymbol{v}_1}(\bar{\boldsymbol{x}}) + \mu\xi_2$. We already checked that before that query, all responses of the oracle are consistent with minimizing $F_{\boldsymbol{A},\boldsymbol{v}_1}$, hence on an event $\mathcal{E}_1$ of probability at least $1/3$, one has $T_2 \geq Q_1$. Next, consider the hypothetical case when at time T_2, the algorithm is also given the information of $\bar{\boldsymbol{x}}$ and is allowed to store this vector. Given this information, optimizing $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$ reduces to optimizing $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$ since we already know that the minimum is achieved within $B_d(\bar{\boldsymbol{x}}, \xi_2/3)$. Further, any query outside of this ball either

- returns a vector $\boldsymbol{v}_{1,i}$ which does not give any useful information for the minimization ($\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are sampled independently and $\bar{\boldsymbol{x}}$ is given),
- or returns a row from $\boldsymbol{A}$, as covered by the original proof.

Hence, on an event $\mathcal{E}_2$ of probability at least $1/3$, even with the extra information of $\bar{\boldsymbol{x}}$, during the next Q_2 queries starting from T_2, the algorithm does not query a $\Theta(\mu\xi_2^3)-$suboptimal solution to $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$. This holds a fortiori for the model when the algorithm is not given $\bar{\boldsymbol{x}}$ at time T_2.

Recursive construction of a p-level class of functions $F_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_p}$

Similarly as in the last section, one can inductively construct the sequence of functions $F_{\boldsymbol{A},\boldsymbol{v}_1}$, $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2}$, $F_{\boldsymbol{A},\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3}$, etc. Formally, the induction is constructed as follows: let $(\boldsymbol{v}_p)_{p\geq 1}$ be an i.i.d. sequence of N i.i.d. vectors $(\boldsymbol{v}_{k,i})_{i\in[N]}$ sampled from the rescaled hypercube $d^{-1/2}\{\pm 1\}^d$. Next, we pose

$$G_{\boldsymbol{A},\boldsymbol{v}_1}(\boldsymbol{x}) = \mu^{(1)} \max_{i\in[N]} \left(\boldsymbol{v}_{1,i}^\top \left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}^{(1)}}{s^{(1)}} \right) - i\gamma \right),$$

where $\mu^{(1)} = \mu$, $\bar{\boldsymbol{x}}^{(1)} = \boldsymbol{0}$ and $s^{(1)} = 1$. For $k \geq 1$, we pose

$$\bar{\boldsymbol{x}}^{(k+1)} = \bar{\boldsymbol{x}}^{(k)} - \frac{s^{(k)}}{2\sqrt{N}} \sum_{i\in[N]} P_{\boldsymbol{A}^\perp}(\boldsymbol{v}_{k,i}), \quad \text{and} \quad F^{(k)} := G_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_k}(\bar{\boldsymbol{x}}^{(k)}) + \mu^{(k)}\xi_{k+1},$$

for a certain parameter ξ_{k+1} to be specified. We then define the next level as

$$G_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_{k+1}}(\boldsymbol{x}) := \max\left\{ G_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_k}(\boldsymbol{x}), G_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_k}(\bar{\boldsymbol{x}}^{(k+1)}) + \frac{\mu^{(k)}\xi_{k+1}}{3} \cdot \right.$$
$$\left. \max\left\{ 1 + \frac{\|\boldsymbol{x}-\bar{\boldsymbol{x}}^{(k+1)}\|_2}{s^{(k)}}, 1 + \frac{\xi_{k+1}}{6} + \frac{\xi_{k+1}}{18} \max_{i\in[N]} \left(\boldsymbol{v}_{k+1,i}^\top \left(\frac{\boldsymbol{x}-\bar{\boldsymbol{x}}^{(k+1)}}{s^{(k)}\xi_{k+1}/9} \right) - i\gamma \right) \right\} \right\}.$$

We then pose $\mu^{(k+1)} := \mu^{(k)}\xi_{k+1}^2/54$ and $s^{(k+1)} := s^{(k)}\xi_{k+1}/9$, which closes the induction. The optimization functions are defined simply as

$$F_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_{k+1}}(\boldsymbol{x}) = \max\left\{ \|\boldsymbol{A}\boldsymbol{x}\|_\infty - \mu, G_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_{k+1}}(\boldsymbol{x}) \right\}.$$

We checked before that we can use $\xi_2 = 1/(16\sqrt{d})$. For general $k \geq 0$, given that the form of the function slightly changes to incorporate the absolute term (see $\tilde{F}_{\boldsymbol{A},\boldsymbol{v}_2}$), this constant may differ slightly. In any case, one has $\xi_k = \Theta(1/\sqrt{d})$. Now fix a construction level $p \geq 1$ and for any $k \in [p]$, let T_k be the first time that a point with function value at most $F^{(k)}$ is queried. For convenience let $T_0 = 0$. Using the same arguments as above recursively, we can show that on an event $\mathcal{E}_k$ with probability at least $1/3$,

$$T_k - T_{k-1} \geq Q_k = Q\left(\Theta\left(\frac{\mu}{s^{(k)}} \right); M, d \right)$$

Next note that the sequence $F^{(k)}$ is decreasing and by construction, if one finds a $\mu^{(p)}\xi_{p+1}$-suboptimal point of $F_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_p}$, then this point has value at most $F^{(p)}$. As a result, for an

algorithm that finds a $\mu^{(p)}\xi_{p+1}$-suboptimal point, the times $T_0, \dots, T_p$ are all well defined and non-decreasing. We recall that $\mu = \Theta(\sqrt{d}\epsilon)$. Therefore, we can still have $\mu/s^{(p)} \leq \sqrt{\epsilon}$ and $\mu^{(p)}\xi_{p+1} \geq \epsilon^2$ for $p = \Theta(\frac{\ln \frac{1}{\epsilon}}{\ln d})$. Combining these observations, we showed that when optimizing the functions $F_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_p}$ to accuracy $\Theta(\mu^{(p)}\xi_{p+1}) = \Omega(\epsilon^2)$, the total number of queries Q satisfies

$$\mathbb{E}[Q] \geq \frac{1}{3}\sum_{k\in[p]} Q_k \geq \frac{p}{3}Q(\sqrt{\epsilon}; M, d) = \Theta\left(\frac{d^{4/3}\ln\frac{1}{\epsilon}}{\ln^{4/3} d}\left(\frac{d\ln\frac{1}{\epsilon}}{M + d\ln d}\right)^{4/3}\right).$$

Changing ϵ to ϵ^2 proves the desired result.

Theorem 8.7. *For $\epsilon \leq 1/d^8$ and any $\delta \in [0,1]$, any (potentially randomized) algorithm guaranteed to minimize 1-Lipschitz convex functions over the unit ball with ϵ accuracy uses at least $d^{1.25-\delta}\ln\frac{1}{\epsilon}$ bits of memory or makes $\tilde{\Omega}(d^{1+4\delta/3}\ln\frac{1}{\epsilon})$ queries.*

The same recursive construction can be applied to the results from Theorems 8.4 and 8.5 to improve their oracle-complexity lower bounds by a factor $\frac{\ln\frac{1}{\epsilon}}{\ln d}$, albeit with added technicalities due to the adaptivity of their class of functions. This yields Theorem 8.2.

8.7 Discussion and Conclusion

To the best of our knowledge, this work is the first to provide some positive trade-off between oracle-complexity and memory-usage for convex optimization or the feasibility problem, as opposed to lower-bound impossibility results from [Mar+22] or Chapter 7. Our trade-offs are more significant in a high accuracy regime: when $\ln\frac{1}{\epsilon} \approx d^c$, for $c > 0$ our trade-offs are polynomial, while the improvements when $\ln\frac{1}{\epsilon} = \text{poly}(\ln d)$ are only in $\ln d$ factors. A natural open direction [WS19] is whether there exist algorithms with polynomial trade-offs in that case. We also show that in the exponential regime $\ln\frac{1}{\epsilon} \geq \Omega(d\ln d)$, gradient descent is not Pareto-optimal. Instead, one can keep the optimal memory and decrease the dependence in ϵ of the oracle-complexity from $\frac{1}{\epsilon^2}$ to $(\ln\frac{1}{\epsilon})^d$. The question of whether the exponential dependence in d is necessary is left open. Last, our algorithms rely on the consistency of the oracle, which allows re-computations. While this is a classical assumption, gradient descent and classical cutting-plane methods do not need it; removing this assumption could be an interesting research direction (potentially, this could also yield stronger lower bounds).

8.8 Appendix: Memory-Constrained Gradient Descent for the Feasibility Problem

In this section, we prove a simple result showing that memory-constrained gradient descent applies to the feasibility problem. We adapt the algorithm described in [WS19].

We now prove that this memory-constrained gradient descent gives the desired result of Proposition 8.1.

Input: Number of iterations T, computation accuracy $\eta \leq 1$, target accuracy $\epsilon \leq 1$

Initialize: $\boldsymbol{x} = \mathbf{0}$;
for $t = 0, \dots, T$ **do**
    Query the oracle at $\boldsymbol{x}$
    if $\boldsymbol{x}$ *successful* **then return** x;
    Receive a separation vector $\boldsymbol{g}$ with accuracy η
    Update $\boldsymbol{x}$ as $\boldsymbol{x} - \epsilon \boldsymbol{g}$ up to accuracy η
end
return $\boldsymbol{x}$

Algorithm 8.11: Memory-constrained gradient descent

Proof of Proposition 8.1 Denote by $\boldsymbol{x}_t$ the state of $\boldsymbol{x}$ at iteration t, and $\boldsymbol{g}_t$ (resp. $\tilde{\boldsymbol{g}}_t$) the separation oracle without rounding errors (resp. with rounding errors) at $\boldsymbol{x}_t$. By construction,

$$\|\boldsymbol{x}_{t+1} - (\boldsymbol{x}_t + \epsilon \tilde{\boldsymbol{g}}_t)\| \leq \eta \quad \text{and} \quad \|\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t\| \leq \eta. \tag{8.11}$$

As a result, recalling that $\|\boldsymbol{g}_t\| = 1$,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq (\|\boldsymbol{x}_t + \epsilon \tilde{\boldsymbol{g}}_t - \boldsymbol{x}^\star\| + \eta)^2 \leq (\|\boldsymbol{x}_t + \epsilon \boldsymbol{g}_t - \boldsymbol{x}^\star\| + (1+\epsilon)\eta)^2 \leq \|\boldsymbol{x}_t + \epsilon \boldsymbol{g}_t - \boldsymbol{x}^\star\|^2 + 20\eta.$$

By assumption, Q contains a ball $B_d(\boldsymbol{x}^\star, \epsilon)$ for $\boldsymbol{x}^\star \in B_d(0, 1)$. Then, because $\boldsymbol{g}_t$ separates $\boldsymbol{x}_t$ from $B_d(\boldsymbol{x}^\star, \epsilon)$, one has $\boldsymbol{g}_t^\top (\boldsymbol{x}^\star - \boldsymbol{x}_t) \geq \epsilon$. Therefore,

$$\begin{aligned} \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 + 2\epsilon \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^\star) + \epsilon^2 \|\boldsymbol{g}_t\|^2 + 20\eta \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \epsilon^2 + 20\eta. \end{aligned}$$

Then, take $\eta = \epsilon^2/40$ and $T = \frac{8}{\epsilon^2}$. If iteration T was performed, we have using the previous equation

$$\|\boldsymbol{x}_T - \boldsymbol{x}^\star\|^2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 - \frac{\epsilon^2}{2} T \leq 4 - \frac{\epsilon^2}{2} T \leq 0.$$

Hence, $\boldsymbol{x}_T$ is an ϵ-suboptimal solution.

We now turn to the memory usage of gradient descent. It only needs to store $\boldsymbol{x}$ and $\boldsymbol{g}$ up to the desired accuracy $\eta = \mathcal{O}(\epsilon^2)$. Hence, this storage and the internal computations can be done with $\mathcal{O}(d \ln \frac{d}{\epsilon})$ memory. Because we suppose that $\epsilon \leq \frac{1}{\sqrt{d}}$, this gives the desired result. ■

Chapter 9

Gradient Descent is Pareto-Optimal in the Oracle Complexity and Memory Trade-off for Feasibility Problems

9.1 Introduction

We consider the *feasibility problem* in which one has access to a separation oracle for a convex set contained in the unit ball $Q \subset B_d(\mathbf{0}, 1) := \{\boldsymbol{x} \in \mathbb{R}^d, \|\boldsymbol{x}\|_2 \leq 1\}$ and aims to find a point $\boldsymbol{x} \in Q$. For the feasibility problem with accuracy ϵ we assume that Q contains a Euclidean ball of radius ϵ. The feasibility problem is arguably one of the most fundamental problems in optimization and mathematical programming. For reference, the feasibility problem is tightly related to the standard non-smooth convex optimization problem [NY83; Nes03] in which one aims to minimize Lipschitz smooth functions having access to a first-order oracle: the gradient information can be used as a separation oracle from the set of minimizers. Both setups have served as key building blocks for numerous other problems in optimization, computer science, and machine learning.

The efficiency of algorithms for feasibility problems is classically measured by either their time complexity or their oracle complexity, that is, the number of calls to the oracles needed to provide a solution. Both have been extensively studied in the literature: in the regime $\epsilon \leq 1/\sqrt{d}$ the textbook answer is that $\Theta(d \ln 1/\epsilon)$ oracle calls are necessary [NY83] and these are achieved by the broad class of cutting plane methods, which build upon the seminal ellipsoid method [YN76a; Kha80]. While the ellipsoid method has suboptimal $\mathcal{O}(d^2 \ln 1/\epsilon)$ oracle complexity, subsequent works including the inscribed ellipsoid [Tar88; Nes89], the celebrated volumetric center or Vaidya's method [AV95; Vai96; Ans97], or random walk based methods [Lev65; BV04] achieve the optimal $\mathcal{O}(d \ln 1/\epsilon)$ oracle complexity. In terms of runtime, in a remarkable tour-de-force [LSW15; Jia+20] improved upon previous best-known $\mathcal{O}(d^{1+\omega} \ln 1/\epsilon)$[1] complexity of Vaidya's method and showed that cutting planes can be implemented with $\mathcal{O}(d^3 \ln 1/\epsilon)$ time complexity.

In practice, however, cutting planes are seldom used for large-dimensional applications and are rather viewed as impractical. While they achieve the optimal oracle complexity, these

[1] $\omega < 2.373$ denotes the exponent of matrix multiplication

typically require storing all previous oracle responses, or at the very least, a summary matrix that uses $\Omega(d^2 \ln 1/\epsilon)$ bits of memory and needs to be updated at each iteration (amortized $\Omega(d^2)$ runtime per iteration). Instead, gradient-descent-based methods are often preferred for their practicality. These only keep in memory a few vectors, hence use only $\mathcal{O}(d \ln 1/\epsilon)$ bits and $\mathcal{O}(d \ln 1/\epsilon)$ runtime per iteration, but require $\mathcal{O}(1/\epsilon^2)$ oracle queries which is largely suboptimal for $\epsilon \ll 1/\sqrt{d}$. These observations, as well as other practical implementation concerns, motivated the study of trade-offs between the oracle complexity and other resources such as memory usage [WS19; Mar+22; CP23] or communication [LLZ20; Red+16; SSZ14; Smi+17; Mot+13; ZDW12; Wan+18; WWS17].

Previous results on oracle complexity/memory trade-offs for convex optimization and feasibility problems. Understanding the trade-offs between oracle complexity and memory usage was first formally posed as a COLT 2019 open problem in [WS19] in the convex optimization setup. Quite surprisingly, in the standard regime $\frac{1}{\sqrt{d}} \geq \epsilon \geq e^{-d^{o(1)}}$, gradient descent and cutting planes are still the only two known algorithms in the oracle complexity/memory trade-off for both the feasibility problem and first-order Lipschitz convex optimization. Other methods have been proposed for other optimization settings to reduce memory usage including limited-memory Broyden– Fletcher– Goldfarb– Shanno (BFGS) methods [Noc80; LN89], conjugate gradient methods [FR64; HZ06], Newton methods variants [PW17; RM19], or custom stepsize schedules [DVR23; AP23], however these do not improve in memory usage or oracle complexity upon gradient descent or cutting-planes in our problem setting. However, for super-exponential regimes, in Chapter 8 we proposed recursive cutting-plane algorithms that provide some trade-off between memory and oracle complexity, and in particular strictly improve over gradient descent whenever $\epsilon \leq e^{-\Omega(d \ln d)}$.

More advances were made on impossibility results. [Mar+22] made the first breakthrough by showing that having both optimal oracle complexity $\mathcal{O}(d \ln 1/\epsilon)$ and optimal $\mathcal{O}(d \ln 1/\epsilon)$-bit memory is impossible: any algorithm either uses $d^{1+\delta}$ memory or makes $\tilde{\Omega}(d^{4/3(1-\delta)})$ oracle calls, for any $\delta \in [0, 1/4]$. This result was then improved in Chapter 7 and in [CP23] to show that having optimal oracle complexity is impossible whenever one has less memory than cutting planes: for $\delta \in [0, 1]$, any deterministic (resp. randomized) first-order convex optimization algorithm uses $d^{1+\delta}$ memory or makes $\tilde{\Omega}(d^{4/3-\delta/3})$ queries (resp. $\Omega(d^{7/6-\delta/6-o(1)})$ queries if $\epsilon \leq e^{-\ln^5 d}$). All these previous query lower bounds were proved for *convex optimization*. For the more general *feasibility problem*, the query lower bound can be further improved to $\tilde{\Omega}(d^{2-\delta})$ for deterministic algorithms as we showed in Chapter 7.

Our contribution. While the previous query lower bounds demonstrated the advantage of having larger memory, their separation in oracle complexity is very mild: the oracle complexity of memory-constrained algorithms is lower bounded to be d times more than the optimal complexity $\mathcal{O}(d \ln 1/\epsilon)$[2]. This significantly contrasts with the oracle complexity $\mathcal{O}(1/\epsilon^2)$ of gradient descent which is arbitrarily suboptimal for small accuracies ϵ. In particular, the question of whether there exists linear-memory algorithms with only logarithmic dependency $\ln 1/\epsilon$ for their oracle complexity remained open. Given that gradient descent is most com-

[2]In fact, the lower bound trade-offs in [Mar+22; CP23] or in Chapter 7 do not include any dependency in ϵ. However, in Chapter 8 we noted that all previous lower bounds can be modified to add the $\ln 1/\epsilon$ factor

monly used and arguably more practical than cutting-planes, understanding whether it can be improved in the oracle complexity/memory trade-off is an important question to address.

In this work, we provide some answers to the following questions for the *feasibility problem*. (1) Can we improve the query complexity of gradient descent while keeping its optimal memory usage? This question was asked as one of the COLT 2019 open problems of [WS19] for convex optimization. (2) What is the dependency in ϵ for the oracle complexity of memory-constrained algorithms? Our first result for deterministic algorithms is summarized below, where $o(1)$ refers to a function of d vanishing as $d \to \infty$.

Theorem 9.1. *Fix $\alpha \in (0, 1]$. Let d be a sufficiently large integer (depending on α) and $\frac{1}{\sqrt{d}} \geq \epsilon \geq e^{-d^{o(1)}}$. Then, for any $\delta \in [0, 1]$, any deterministic algorithm solving feasibility problems up to accuracy ϵ either uses $M = d^{1+\delta}$ bits of memory or makes at least $1/\left(d^{\alpha\delta}\epsilon^{2\cdot\frac{1-\delta}{1+(1+\alpha)\delta}-o(1)}\right)$ queries.*

For randomized algorithms, we have the following.

Theorem 9.2. *Let d be a sufficiently large integer and $\frac{1}{\sqrt{d}} \geq \epsilon \geq e^{-d^{o(1)}}$. Then, for any $\delta \in [0, 1/4]$, any randomized algorithm solving feasibility problems up to accuracy ϵ with probability at least $\frac{9}{10}$ either uses $M = d^{1+\delta}$ bits of memory or makes at least $1/\left(d^{2\delta}\epsilon^{2(1-4\delta)-o(1)}\right)$ queries.*

Fig. 9.1 provides a visualization of these trade-offs and a comparison to previous results. As a result, gradient descent is Pareto-optimal (up to a factor d) in the trade-off between oracle complexity and memory usage for the feasibility problem. In other words, having optimal memory usage requires suffering the worst-case dependency $1/\epsilon^2$ for the oracle complexity. Further, our results also reveal a sharp phase transition for the oracle complexity of memory-constrained algorithms: deterministic algorithms that have less than quadratic memory in the dimension d suffer a factor polynomial in $1/\epsilon$ in their oracle complexity. In contrast, we recall that using quadratic memory $\mathcal{O}(d^2 \ln 1/\epsilon)$, cutting plane methods achieve a logarithmic dependency in $1/\epsilon$ for the oracle complexity: $\mathcal{O}(d \ln 1/\epsilon)$. For randomized algorithms, however, our result only implies that if an algorithm has less than $d^{5/4}$ memory it suffers a factor polynomial in $1/\epsilon$ in their oracle complexity.

We stress that since the feasibility problem generalizes the convex optimization setup, lower bound trade-offs for feasibility problems do not imply lower bound trade-offs for convex optimization, while the converse holds. We hope, however, that the techniques developed in this work can lead to future results for convex optimization.

On the tightness of the results. Theorems 9.1 and 9.2 are simplified versions of the more precise lower bound results in Theorems 9.8 and 9.12 respectively. These explicit the term $o(1)$ in the exponent of our lower bounds, which is of order $\frac{\ln\ln d \vee \ln\frac{\ln(1/\epsilon)}{\ln d}}{\ln d}$, and also provide lower bound trade-offs in exponential regimes when $\epsilon = e^{-d^c}$, that degrade gracefully as $c > 0$ grows. While our lower bounds are likely not perfectly tight, our results in Chapter 8 showed that some dependency in $\ln\frac{\ln(1/\epsilon)}{\ln d}/\ln d$ in the query lower bound exponents is necessary, since they provide memory-constrained algorithms for exponential regimes. In particular, when

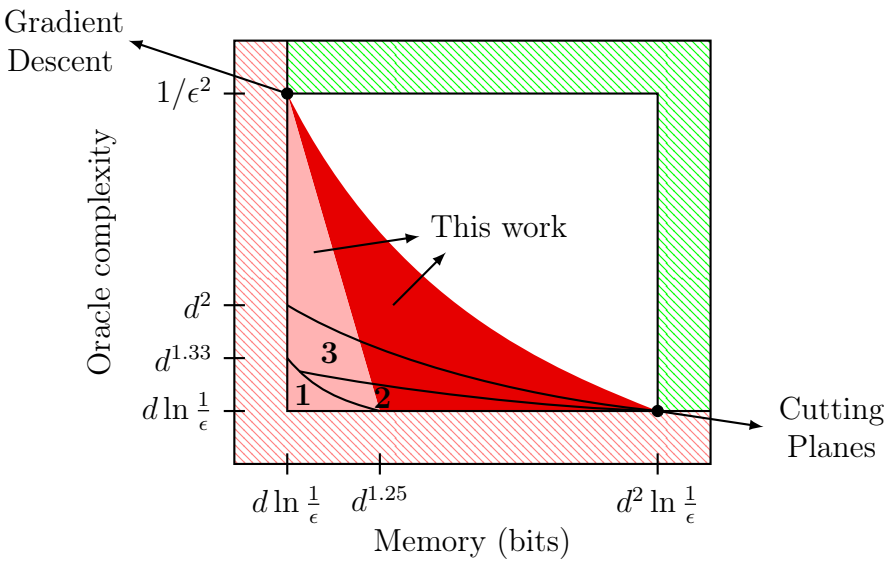

Figure 9.1: Trade-offs between available memory and oracle complexity for the feasibility problem with accuracy ϵ in dimension d, in the regime $\frac{1}{\sqrt{d}} \geq \epsilon \geq e^{-d^{o(1)}}$ (adapted from [WS19]). The dashed pink (resp. green) region corresponds to historical information-theoretic lower bounds (resp. upper bounds). The region **1** and **2** correspond to the lower bound trade-offs from [Mar+22] and [CP23] respectively for randomized algorithms. The region **3** corresponds to the lower bound from Chapter 7 for deterministic algorithms. In this work, we show that the red (resp. pink) solid region is not achievable for deterministic (resp. randomized) algorithms.

$\epsilon \leq e^{-\Omega(d \ln d)}$ we achieved query complexity $(\mathcal{O}(\ln 1/\epsilon))^d$ with the optimal $\mathcal{O}(d \ln 1/\epsilon)$-bit memory. Hence in this regime, lower bounds cannot be polynomial in $1/\epsilon$ anymore.

The bounds for the deterministic case depend on a parameter $\alpha \in (0, 1]$. An inspection of the complete bound from Theorem 9.8 shows that in Theorem 9.1 one can take $\alpha = \omega(\ln \ln d / \ln d)$. This term is due to our analysis of a probing game described in Section 9.3. Improving the bounds for this game to delete this term may be possible; we refer to Section 9.3 for more details.

Additional works on learning with memory constraints. The impact of memory constraints on learning problems has received much attention in the past years. For parity learning, [Raz17] first proved that an exponential number of queries is necessary if the memory is sub-quadratic. Similar results were then obtained for other parity problems [KRT17; MM17; Raz18; MM18; BOY18; GRT18; GRT19; Gar+21]. Trade-offs between memory usage and sample complexity were also studied for linear regression [SD15; SSV19], principal component analysis (PCA) [MCJ13], learning under the statistical query model [SVW16], minimizing regret in online learning [PZ23; PR23; Sri+22] and other general learning problems [Bro+21; BBS22].

9.1.1 Outline of the chapter

We formalize the setup and give preliminary definitions in Section 9.2 then give an overview of the proof structure and main technical contributions in Section 9.3. We mainly give details for the deterministic case (Theorem 9.1) but discuss additional proof components for the randomized case (Theorem 9.2) in Section 9.3.4. We then prove our lower bound trade-off for deterministic algorithms in Section 9.4 and for randomized algorithms in Section 9.5.

9.2 Formal Setup and Notations

We aim to study the trade-off between memory usage and query complexity for the feasibility problem over the unit ball $B_d(\mathbf{0}, 1)$. The goal is to find an element $\boldsymbol{x} \in Q$ for a convex set $Q \subset B_d(\mathbf{0}, 1)$ having access to a separation oracle $\mathcal{O}_Q : B_d(\mathbf{0}, 1) \to \mathbb{R}^d \cup \{\mathsf{Success}\}$ such that for any query $\boldsymbol{x} \in B_d(\mathbf{0}, 1)$, the oracle either returns $\mathsf{Success}$ if $\boldsymbol{x} \in Q$ or provides a separating hyperplane $\boldsymbol{g} \in \mathbb{R}^d$ for Q, i.e., such that

$$\forall \boldsymbol{x}' \in Q, \quad \boldsymbol{g}^\top (\boldsymbol{x}' - \boldsymbol{x}) < 0.$$

The oracle is allowed to be randomized and potentially iteration-dependent (sometimes referred to as oblivious). For the feasibility problem with accuracy $\epsilon > 0$, we assume that the set Q contains a ball of radius ϵ and ϵ is known to the algorithm. We note that other works sometimes consider a stronger version of the feasibility problem in which the algorithm either finds a feasible point $\boldsymbol{x} \in Q$ or proves that Q does not contain a ball of size ϵ [LSW15; Jia+20]. Our impossibility results hence also apply to this setting as well.

We next formally define M-bit memory-constrained algorithms given a query space $\mathcal{S}$ and a response space $\mathcal{R}$. Intuitively, these can only store M bits of memory between each oracle call.

Definition 9.1 (Memory-constrained algorithm)**.** *An M-bit memory-constrained deterministic algorithm is specified by query functions $\psi_{query,t} : \{0,1\}^M \to \mathcal{S}$ and update functions $\psi_{update,t} : \{0,1\}^M \times \mathcal{S} \times \mathcal{R} \to \{0,1\}^M$ for $t \geq 1$. At the beginning of the feasibility problem, the algorithm starts with the memory state $\mathsf{Memory}_0 = 0_M$. At each iteration $t \geq 1$, it makes the query $\boldsymbol{x}_t = \psi_{query,t}(\mathsf{Memory}_{t-1})$, receives a response $r_t \in \mathcal{R}$ from a separation oracle, then updates its memory state $\mathsf{Memory}_t = \psi_{update,t}(\mathsf{Memory}_{t-1}, \boldsymbol{x}_t, r_t)$.*

For M-bit memory-constrained randomized algorithms, the query and update functions can additionally use fresh randomness at every iteration.

For the feasibility problem, we have in particular $\mathcal{S} = B_d(\mathbf{0}, 1)$ and $\mathcal{R} = \mathbb{R}^d \cup \{\mathsf{Success}\}$. Note that the defined notion of memory constraint is quite mild. Indeed, the algorithm is not memory-constrained for the computation of the query and update functions and can potentially use unlimited memory and randomness for these; the constraint only applies *between* iterations. A fortiori, our lower bounds also apply to stronger notions of memory constraints.

Notations. For any $\boldsymbol{x} \in \mathbb{R}^d$ and $r \geq 0$, we denote by $B_d(\boldsymbol{x}, r) = \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \leq 1\}$ the ball centered at $\boldsymbol{x}$ and of radius r. We denote the unit sphere by $S_{d-1} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1\}$ By default, the norm $\|\cdot\|$ refers to the Euclidean norm. For any integer $n \geq 1$, we use the shorthand $[n] = \{1, \dots, n\}$. We denote $\boldsymbol{e} := (1, 0, \dots, 0) \in \mathbb{R}^d$ and write $\boldsymbol{I}_d$ for the identity matrix in $\mathbb{R}^d$. We use the notation $Span(\cdot)$ to denote the subspace spanned by considered vectors or subspaces. For a subspace E, Proj_E denotes the orthogonal projection onto E. For a finite set S, we denote by $\mathcal{U}(S)$ the uniform distribution over S. Last, for $d \geq k \geq 1$, to simplify the wording, a randomi uniform k-dimensional subspace of $\mathbb{R}^d$ always refers to a k-dimensional subspace sampled according to the normalized Haar measure over the Grassmannian $\boldsymbol{Gr}_k(\mathbb{R}^d)$.

9.3 Technical Overview of the Proofs

In this section, we mainly give an overview of the proof of Theorem 9.1. The result for randomized algorithms Theorem 9.2 follows the same structure with a few differences discussed in Section 9.3.4.

9.3.1 Challenges for having ϵ-dependent query lower bounds

We first start by discussing the challenges in improving the lower bounds from previous works [Mar+22; CP23] or from Chapter 7. To make our discussion more concrete, we use as an example the construction of [Mar+22] who first introduced proof techniques to obtain query complexity/memory lower bounds, and explain the challenges to extend their construction and have stronger lower bounds. The constructions from the subsequent work from Chapter 7 or [CP23] use very similar classes of functions and hence present similar challenges. [Mar+22] defined the following hard class of functions to optimize

$$F_{\boldsymbol{A},\boldsymbol{v}_1,\dots,\boldsymbol{v}_N}(\boldsymbol{x}) := \max\left\{\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta, \delta\left(\max_{i \leq N} \boldsymbol{v}_i^\top \boldsymbol{x} - i\gamma\right)\right\}, \tag{9.1}$$

where $\boldsymbol{A} \sim \mathcal{U}(\{\pm 1\}^{d/2 \times d})$ is sampled with i.i.d. binary entries, the vectors $\boldsymbol{v}_i \overset{i.i.d.}{\sim} \mathcal{U}(\frac{1}{\sqrt{d}}\{\pm 1\}^d)$ are sampled i.i.d. from the rescaled hypercube, and $\gamma, \delta, \eta \ll 1$ are fixed parameters. We briefly give some intuition for why this class is hard to optimize for memory-constrained algorithms.

The first term $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta$ acts as a barrier wall term: in order to observe the second term of the function which has been scaled by a small constant δ, one needs to query vectors $\boldsymbol{x}$ that are approximately orthogonal to $\boldsymbol{A}$. On the other hand, the second term is a Nemirovski function [Nem94; BS18; Bub+19] that was used for lower bounds in parallel optimization and enforces the following behavior: with high probability an optimization algorithm observes the vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ in this order and needs to query "robustly-independent" queries to observe these. A major step of the proof is then to show that finding many queries that are (1) approximately orthogonal to $\boldsymbol{A}$ and (2) robustly-independent, requires re-querying $\Omega(d)$ of the rows of $\boldsymbol{A}$. This is done by proving query lower bounds on an Orthogonal Vector Game that simulates this process. At the high level, one only receives subgradients that

are rows of $\boldsymbol{A}$ *one at a time*, while findings vectors that are approximately orthogonal to $\boldsymbol{A}$ requires information on *all* of its rows.

A possible hope to improve the query lower bounds is to repeat this construction recursively at different depths, for example sampling other matrices $\boldsymbol{A}_p$ for $p \in [P]$ and adding a term $\delta^{(p)}(\|\boldsymbol{A}_p\boldsymbol{x}\|_\infty - \eta^{(p)})$ to Eq (9.1). This gives rise to a few major challenges. First, for the recursive argument to work, one needs to ensure that the algorithm has to explore many robustly-independent queries at each depth. While this is ensured by the Nemirovski function (because it guides the queries in the direction of vectors $-\boldsymbol{v}_i$ sequentially) for the last layer, this is not true for any term of the form $\|\boldsymbol{A}\boldsymbol{x}\|_\infty - \eta$: a randomly generated low-dimensional subspace can easily observe all rows of $\boldsymbol{A}$ through subgradients. Intuitively, a random l dimensional subspace cuts a hypercube $[-1,1]^d$ through all faces even if $l \ll d$ (potentially logarithmic in d). Conversely, the Nemirovski function does not enforce queries to be within a nullspace of a large incompressible random matrix. Further, once the vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ have been observed, querying them again poses no difficulty (without having to query robustly-independent vectors): assuming we can find a point $\hat{\boldsymbol{x}}$ at the intersection of the affine maps for which $\boldsymbol{v}_i^\top \hat{\boldsymbol{x}} - i\gamma$ is roughly equal for all $i \in [N]$, it suffices to randomly query points in the close neighborhood of $\hat{\boldsymbol{x}}$ to observe all vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ again.

In our proof, we still use a recursive argument to give lower bounds and build upon these proof techniques, however, the construction of the hard instances will need to be significantly modified.

9.3.2 Construction of the hard class of feasibility problems

The construction uses $P \geq 2$ layers and our goal is to show that solving the constructed feasibility problems requires an exponential number of queries in this depth P. We sample P i.i.d. uniform $\tilde{d}$-dimensional linear subspaces $E_1, \ldots, E_P$ of $\mathbb{R}^d$ where $\tilde{d} = \lceil d/(2P) \rceil$. The feasible set is defined for some parameters $0 < \delta_1 < \ldots < \delta_P$ via

$$Q_{E_1,\ldots,E_P} := B_d(\boldsymbol{0}, 1) \cap \left\{ \boldsymbol{x} : \boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2} \right\} \cap \bigcap_{p \in [P]} \left\{ \boldsymbol{x} : \|\mathrm{Proj}_{E_p}(\boldsymbol{x})\| \leq \delta_p \right\}. \tag{9.2}$$

Hence, the goal of the algorithm is to query vectors approximately perpendicular to all subspaces $E_1, \ldots, E_P$. The first constraint $\boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2}$ is included only to ensure that the algorithm does not query vectors with small norm. The parameters δ_p are chosen to be exponentially small in $P - p$ via $\delta_p = \delta_P / \mu^{P-p}$ for a factor $1 \leq \mu = \mathcal{O}(d^{3/2})$. The choice of the factor μ is crucial: it needs to be chosen as small as possible to maximize the number of layers P in the construction while still emulating a feasibility problem with given accuracy ϵ.

We now introduce the procedure to construct our separation oracles, which are designed to reveal information on subspaces E_p with smallest possible index $p \in [P]$. For a sequence of linear subspaces $V_1, \ldots, V_r$ of $\mathbb{R}^d$ that we refer to as *probing* subspaces, we introduce the following function which roughly serves as a separation oracle for $Span(V_1, \ldots, V_r)^\perp$.

$$\boldsymbol{g}_{V_1,\ldots,V_r}(\boldsymbol{x}; \delta) := \begin{cases} \frac{\mathrm{Proj}_V(\boldsymbol{x})}{\|\mathrm{Proj}_V(\boldsymbol{x})\|} & \text{if } \|\mathrm{Proj}_V(\boldsymbol{x})\| > \delta, \\ \mathsf{Success} & \text{otherwise,} \end{cases} \quad \text{where } V = Span(V_i, i \in [r]). \tag{9.3}$$

In practice, all probing subspaces will have the same fixed dimension $l = \mathcal{O}(\ln d)$, except for the last layer for $p = P$ for which we need to finely tune this dimension parameter l_P for the tightness of our results. We ignore this detail in the present overview. We introduce an adaptive feasibility procedure (see Procedure 9.3) with the following structure. For each level $p \in [P]$ we keep some probing subspaces $V_1^{(p)}, V_2^{(p)}, \ldots$. These are sampled uniformly within l-dimensional subspaces of E_p and are designed to "probe" for the query being approximately perpendicular to the complete space E_p. Because they have dimension $l = \mathcal{O}(\ln d) \ll \tilde{d}$, they each reveal little information about E_p. On the other hand, they may not perfectly probe for the query being perpendicular to E_p since they have much lower dimension than E_p. Fortunately, we show these fail only with small probability for adequate choice of parameters (see Lemma 9.2).

Before describing the procedure, we define *exploratory* queries at some depth $p \in [P]$. These are denoted $\boldsymbol{y}_i^{(p)}$ for $i \in [n_p]$ where n_p is the number of current depth-p exploratory queries.

Definition 9.2 (Exploratory queries)**.** *Given previous exploratory queries $\boldsymbol{y}_i^{(p)}$ and probing subspaces $V_i^{(p)}$ for $p \in [P], i \in [n_p]$, we say that $\boldsymbol{x} \in B_d(\mathbf{0}, 1)$ is a depth-p* exploratory *query if:*

1. $\boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2}$,
2. *the query passed all probes from levels $q \leq p$, that is $\boldsymbol{g}_{V_1^{(q)},\ldots,V_{n_q}^{(q)}}(\boldsymbol{x}; \delta_q) = \mathsf{Success}$. Equivalently,*
$$\|Proj_{Span(V_i^{(q)}, i \in [n_q])}(\boldsymbol{x})\| \leq \delta_q, \quad q \in [p] \tag{9.4}$$
3. *and it is robustly-independent from all previous depth-p exploratory queries,*
$$\|Proj_{Span(\boldsymbol{y}_r^{(p)}, r \leq n_p)^\perp}(\boldsymbol{x})\| \geq \delta_p. \tag{9.5}$$

The probing subspaces are updated throughout the procedure. Whenever a new depth-p exploratory query is made, we sample a new l-dimensional linear subspace $V_{n_p+1}^{(p)}$ uniformly in E_p unless there were already $n_p = k$ such subspaces for some fixed parameter $k \in [\tilde{d}]$. In that case, we reset all exploratory queries at depth depths $q \leq p$, pose $n_q = 1$, and sample new subspaces $V_1^{(q)}$ for $q \in [p]$. For each level p, denoting by $\boldsymbol{V}^{(p)} := (V_1^{(p)}, \ldots, V_{n_p}^{(p)})$ the list of current depth-p probing subspaces, we define the oracle as follows

$$\mathcal{O}_{\boldsymbol{V}^{(1)},\ldots,\boldsymbol{V}^{(P)}}(\boldsymbol{x}) := \begin{cases} \boldsymbol{e} & \text{if } \boldsymbol{e}^\top \boldsymbol{x} > -\frac{1}{2} \\ \boldsymbol{g}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \delta_p) & \text{if } \boldsymbol{g}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \delta_p) \neq \mathsf{Success} \\ & \text{and } p = \min\{q \in [P], \boldsymbol{g}_{\boldsymbol{V}^{(q)}}(\boldsymbol{x}; \delta_q) \neq \mathsf{Success}\} \\ \mathsf{Success} & \text{otherwise.} \end{cases} \tag{9.6}$$

The final procedure first tries to use the above oracle, then turns to some arbitrary separation oracle for $Q_{E_1,\ldots,E_P}$ when the previous oracle returns $\mathsf{Success}$. When the procedure returns a vector of the form $\boldsymbol{g}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \delta_p)$ for some $p \in [P]$, we say that $\boldsymbol{x}$ was a depth-p query. We can then check that with high probability, this procedure forms a valid adaptive feasibility problem for accuracy $\epsilon = \delta_1/2$ (Lemma 9.1).

9.3.3 Structure of the proof for deterministic algorithms.

For all $p \in [P]$, we refer to a depth-p period as the interval of time between two consecutive times when the depth-p probing subspaces were reset. We first introduce a depth-p game (Game 9.4) that emulates the run of the procedure for a given depth-p period, the main difference being that the goal of the algorithm is not to query a feasible point in $Q_{E_1,\dots,E_P}$ anymore, but to make k depth-p exploratory queries. This makes the problem more symmetric in the number of layers $p \in [P]$ which will help for a recursive query lower bound argument.

Properties of probing subspaces. We show that the depth-p probing subspaces are a good proxy for testing orthogonality to E_p. This is formalized with the notion of *proper* periods (see Definition 9.3) during which if the algorithm performed a depth-p' query $\boldsymbol{x}_t$ with $p' > p$—hence passed the probes at level p—it satisfies $\|\mathrm{Proj}_{E_p}(\boldsymbol{x}_t)\| \leq \eta_p$ for some parameter $\eta_p \geq \delta_p$ as small as possible. We show in Lemma 9.2 that periods are indeed proper with high probability if we take $\eta_p = \Omega(\sqrt{\tilde{d}k^\alpha}\delta_p)$ for any fixed $\alpha \in (0, 1]$. The proof of this result is one of the main technicalities of the present work and uses a reduction to the following Probing Game.

Input: dimension d, response dimension l, number of exploratory queries k, objective $\rho > 0$

Oracle: Sample independent uniform l-dimensional subspaces $V_1, \dots, V_k$ in $\mathbb{R}^d$.
for $i \in [k]$ **do**
  Player: Based on responses $V_j, j < i$, submit query $\boldsymbol{y}_i \in \mathbb{R}^d$
  Oracle: **return** V_i to the player
end
Player wins if for any $i \in [k]$ there exists a vector $\boldsymbol{z} \in Span(\boldsymbol{y}_j, j \in [i])$ such that

$$\|\mathrm{Proj}_{Span(V_j, j\in[i])}(\boldsymbol{z})\| \leq \rho \quad \text{and} \quad \|\boldsymbol{z}\| = 1.$$

Game 9.1: Probing game

Our goal is to show that no player can win at this game with reasonable probability. In this game, the player needs to output k robustly-independent queries that are perpendicular to the probing spaces. Because these only span at most k dimensions, if the probing subspaces had dimension $l = \Omega(k)$ we could easily prove the desired result (from high-dimensional concentration results akin to the Johnson–Lindenstrauss lemma). This is prohibitive for the tightness of our results, however. In fact, for our result for randomized algorithms, we are constrained to this suboptimal choice of parameters, which is one of the reasons why the lower bound trade-off from Theorem 9.2 does not extend to full quadratic memory d^2. Instead, we show that $l = \mathcal{O}(\ln d)$ are sufficient to ensure that Game 9.1 is impossible with high probability (Theorem 9.5), which is used for the deterministic case. This requires the following result for adaptive random matrices.

Theorem 9.3. *Let $C \geq 2$ be an integer and $m = Cn$. Let $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ be a random matrix such that all coordinates $M_{i,j}$ in the upper-triangle $j > (i-1)C$ are together i.i.d. Gaussians $\mathcal{N}(0,1)$. Further, suppose that for any $i \in [n]$, the Gaussian components $M_{u,v}$ for $v > (u-1)C$ with $v > C(i-1)$ are together independent from the sub-matrix $(M_{u,v})_{u \in [i], v \in [C(i-1)]}$. Then for any $\alpha \in (0,1]$, if $C \geq C_\alpha \ln n$, we have*

$$\mathbb{P}\left(\sigma_1(\boldsymbol{M}) < \frac{1}{6}\sqrt{\frac{C}{n^\alpha}}\right) \leq 3e^{-C/16}.$$

Here $C_\alpha = (C_2/\alpha)^{\ln 2/\alpha}$ for some universal constant $C_2 \geq 8$.

As a remark, even for the case when the matrix is exactly upper-triangular, that is, $M_{i,j} = 0$ for all $j \leq (i-1)C$, we are not aware of probabilistic lower bounds on the smallest singular value. Note that this matrix is upper-triangular on a rectangle instead of a square, which is non-standard. We state the corresponding result for upper-triangular rectangular matrices in Corollary 9.1, which may be of independent interest. For square matrices, previous works showed that the smallest singular value is exponentially small in the dimension n [VT98], which is prohibitive for our purposes. Alternatively, [RZ16] give bounds when the i.i.d. Gaussian part of the matrix is broadly connected, a notion similar to graph expansion properties. However, these assumptions are not satisfied if that Gaussian part corresponds to the upper triangle, for which some nodes are sparsely connected. Further, here we potentially allow the coordinates in the lower-triangle to be adaptive in a subset of coordinates in the upper-triangle. We are not aware of prior works that give singular values lower bounds for adaptive components on non-i.i.d. parts, as opposed to simply deterministic components as considered in [RZ16].

We briefly discuss the tightness of Theorem 9.3. We suspect that the extra factor n^α in the denominator for the bound of $\sigma_1(\boldsymbol{M})$ may be superfluous. One can easily check that the best bound one could hope for here is $\sigma_1(\boldsymbol{M}) = \Omega(\sqrt{C})$. This extra factor n^α is the reason for the term α in the query lower bound from Theorem 9.1, hence shaving off this factor would directly improve our lower bounds trade-offs. The success probability (exponentially small in C, but not in n) is however tight, and experimentally it seems that having $C = \Omega(\ln n)$ is also necessary.

Query lower bounds for an Orthogonal Subspace Game. In the construction of the adaptive feasibility procedure (Procedure 9.3) we reset all exploratory queries and probing subspaces for depths $p' \leq p$ whenever k depth-p queries are performed. This gives a nested structure to periods: a depth-p period is a union of depth-$(p-1)$ periods. We then show that we can reduce the run of a depth-p period to the following Orthogonal Subspace Game 9.2 for appropriate choice of parameters, provided that all contained depth-$(p-1)$ periods are proper. This game is heavily inspired by the Orthogonal Vector Game introduced in [Mar+22].

We prove a query lower bound $\Omega(d)$ for this game if the player does not have sufficient memory and needs to find too many robustly-independent vectors.

Input: dimensions d, $\tilde{d}$; memory M; number of robustly-independent vectors k; number of queries m; parameters β, γ

Oracle: Sample a uniform $\tilde{d}$-dimensional subspace E in $\mathbb{R}^d$ and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \overset{i.i.d.}{\sim} \mathcal{U}(S_d \cap E)$

Player: Observe E and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$, and store an M-bit message Message about these

Oracle: Send samples $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ to player

Player: Based on Message and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ only, return unit norm vectors $\boldsymbol{y}_1, \dots, \boldsymbol{y}_k$

The player wins if for all $i \in [k]$

1. $\|\mathrm{Proj}_E(\boldsymbol{y}_i)\| \le \beta$
2. $\|\mathrm{Proj}_{Span(\boldsymbol{y}_1,\dots,\boldsymbol{y}_{i-1})^\perp}(\boldsymbol{y}_i)\| \ge \gamma$.

Game 9.2: Orthogonal Subspace Game

Theorem 9.4. *Let $d \ge 8$, $C \ge 1$, and $0 < \beta, \gamma \le 1$ such that $\gamma/\beta \ge 12\sqrt{kd/\tilde{d}}$. Suppose that $\frac{\tilde{d}}{4} \ge k \ge 50C\frac{M+2}{\tilde{d}} \ln \frac{\sqrt{d}}{\gamma}$. If the player wins the Orthogonal Subspace Game 9.2 with probability at least $1/C$, then $m \ge \frac{\tilde{d}}{2}$.*

For our lower bounds to reach the query complexity $1/\epsilon^2$ of gradient descent, we need the parameter choices of Theorem 9.4 to be tight. In particular, the robustly-independent parameter γ is allowed to be roughly of the same order of magnitude as the orthogonality parameter δ. Previous works required at least a factor $\sqrt{d}$ between these two parameters. For intuition, having the constraint $\gamma/\beta \ge d^c$ for some constant $c \ge 0$ would at best yield a lower bound query complexity of $1/\epsilon^{2/(1+2c)}$ even for linear memory d.

Reduction from the feasibility procedure to the Orthogonal Subspace Game. We briefly explain the reduction from running a depth-p period of the feasibility game to the Orthogonal Subspace Game 9.2. This will also clarify why query lower bounds heavily rely on the constraint for γ/β from Theorem 9.4. In this context, finding exploratory queries translates into finding robustly-independent vectors (as in Eq (9.5)). Further, provided that the corresponding depth-$(p-1)$ periods are proper, exploratory queries also need to be orthogonal to the subspace E_{p-1} up to the parameter η_{p-1}. In turn, we show that this gives a strategy for the Orthogonal Subspace Game 9.2 for parameters $(\beta, \gamma) = (2\eta_{p-1}, \delta_p)$. Hence, a lower bound on γ/β directly induces a lower bound on the factor parameter $\mu = \delta_{p-1}/\delta_p$ on which the parameter δ_1 depends exponentially.

Also, because the procedure is adaptive, responses for depth-p' queries with $p' > p$ may reveal information about the subspace E_p—this also needs to be taken into account by the reduction. Indeed, although the subspaces E_p and $E_{p'}$ are independent, depth-p' responses are constructed from the depth-p' probing subspaces which are added adaptively on previous depth-p' exploratory queries (see Section 9.3.2), for which the algorithm can use any information it previously had on E_p. However, we show that this information leakage is mild and can be absorbed into a larger memory for the Orthogonal Subspace Game 9.2.

Recursive query lower bounds. The query lower bound for Theorem 9.4 then implies that the algorithm must complete many depth-$(p-1)$ periods within a depth-p period (Lemma 9.6). By simulating one of these depth-$(p-1)$ periods, we show that from a strategy for depth-p periods that performs T_p queries we can construct a strategy for depth-$(p-1)$ periods that uses at most $T_{p-1} \approx \frac{lk}{d} T_p$ (Lemma 9.7). Applying this result recursively reduces the number of allowed queries exponentially with the depth P. Selecting the parameters k and P appropriately gives the query lower bound from Theorem 9.1 for memory-constrained algorithms.

9.3.4 Other proof components for randomized algorithms

For randomized algorithms we cannot construct an adaptive feasibility procedure. In particular, we cannot add a new probing subspace whenever a new exploratory query was performed. Instead, we construct an oblivious iteration-dependent oracle and resample probing subspaces regularly (see formal definition in Eq (9.37)) hoping that for most periods the algorithm did not have time to perform k exploratory queries.

This has a few main implications. First, we cannot use the convenient dimension $l = \mathcal{O}(\ln d)$ for probing subspaces anymore because this heavily relied on the structure of the Probing Game 9.1, for which we sample a new probing subspace just after every new direction is explored. Instead, the probing subspaces $V_1^{(p)}, \dots, V_k^{(p)}$ need to be present at all times in the oracle and kept constant within a depth-p period. Second, we still need to ensure that the algorithm discovers the probing subspaces in the exact order $V_1^{(p)}, \dots, V_k^{(p)}$. Hence we use for each one of these a slightly different orthogonality tolerance parameter. Precisely, instead of using the oracle $\boldsymbol{g}_{V_1,\dots,V_k}$ as in the deterministic case, we define

$$\mathcal{I}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}) = \left\{ i \in [k] : \|\mathrm{Proj}_{V_i}(\boldsymbol{x})\| > \delta_i \right\}.$$

for some parameter $\boldsymbol{\delta} = (\delta_1 > \dots > \delta_k)$ and use the following oracle

$$\tilde{\boldsymbol{g}}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}) := \begin{cases} \frac{\mathrm{Proj}_{V_i}(\boldsymbol{x})}{\|\mathrm{Proj}_{V_i}(\boldsymbol{x})\|} & \text{if } \mathcal{I}_{V_1,\dots,V_k}(\boldsymbol{x}; \boldsymbol{\delta}) \neq \emptyset \text{ and } i = \min \mathcal{I}_{V_1,\dots,V_k}(\boldsymbol{x}; \boldsymbol{\delta}), \\ \mathsf{Success} & \text{otherwise.} \end{cases}$$

The main idea is that while the algorithm does not query vectors slightly orthogonal to V_1, it cannot see V_2, and so on. Unfortunately, concentration bounds only ensure that this is true if the dimension l of the probing subspaces is sufficiently large. In practice, we need $l = \Omega(k \ln d)$. Third and last, as it turns out, we cannot ensure that for all depth-$(p-1)$ periods within a given depth-p period, the probing subspaces were good proxies for being orthogonal to E_{p-1}. In fact, this will most of the time be false. However, given the index of the depth-$(p-1)$ that was improper, we show that the algorithm does perform k exploratory queries during this period. This still gives the desired recursive argument at the expense of giving the algorithm some extra power to select which period to play. As a consequence, the probing subspaces in the played period are not distributed uniformly among l-dimensional subspaces of E_p anymore. To resolve this issue, we adapt the Orthogonal Subspace Game 9.2 to include this additional degree of liberty of the player (see Game 9.14) and show that the query lower bounds still hold (Theorem 9.9).

9.4 Query Complexity / Memory Trade-offs for Deterministic Algorithms

9.4.1 Definition of the feasibility procedure

We give here the detailed construction for the hard class of feasibility problems. We mainly specify the parameters that were already introduced in Section 9.3.2. Fix the parameter $P \geq 2$ for the depth of the nested construction. We assume throughout this chapter that $d \geq 40P$. We sample P uniformly random $\tilde{d}$-dimensional linear subspaces E_p of $\mathbb{R}^d$ where $d := \lfloor d/(2P) \rfloor$. Note that with our choice of parameters ($d \geq 40P$) we have in particular $\tilde{d} \geq \frac{d}{3P}$. We next introduce a parameter $k \in [\tilde{d}]$ and fix $\alpha \in (0,1)$ a constant that will define the sharpness of the Pareto-frontier (the smaller α, the stronger the lower bounds, but these would apply for larger dimensions). Next, we define

$$l := \left\lceil 16 \ln(32 d^2 P) \vee C_\alpha \ln k \right\rceil, \tag{9.7}$$

for some constant $C_\alpha = (\mathcal{O}(1/\alpha))^{\ln 2/\alpha}$ that only depends on α and that will formally be introduced in Theorem 9.3. For the last layer P, we use an extra-parameter $l_P \in [\tilde{d}]$ such that $l_P \geq l$. For convenience, we then define $l_p = l$ for all $p \in [P-1]$. It remains to define some parameters η_p for $p \in [P]$ that will quantify the precision needed for the algorithm at depth p. For the tightness of our results, we need to define the last layer P with different parameters, as follows:

$$\eta_P := \frac{1}{10}\sqrt{\frac{\tilde{d}}{d}} \quad \text{and} \quad \mu_P := 600\sqrt{\frac{dk^{1+\alpha}}{l_P}}.$$

For $p \in [P-1]$, we let

$$\eta_p := \frac{\eta_P/\mu_P}{\mu^{P-p-1}} \quad \text{where} \quad \mu := 600\sqrt{\frac{dk^{1+\alpha}}{l}}. \tag{9.8}$$

The orthogonality parameters $\delta_1, \dots, \delta_P$ are then defined as

$$\delta_p := \frac{\eta_p}{36}\sqrt{\frac{l_p}{\tilde{d}k^\alpha}}, \quad p \in [P-1]. \tag{9.9}$$

Note that at this point, the three main remaining parameters are the depth P, the dimension l_P at the last layer, and k which will serve as the maximum number of exploratory queries within a period. Also, note that if $l_P = l$, then the last layer for $p = P$ is constructed identically as the other ones.

Given these parameters, the feasible set $Q_{E_1,\dots,E_P}$ is given as in Eq (9.2). For any probing subspaces $V_1, \dots, V_r$, we recall the form of the function $\boldsymbol{g}_{V_1,\dots,V_r}(\cdot;\delta)$ from Eq (9.3). Next, we recall the definition of depth-p exploratory queries for $p \in [P]$ from Definition 9.2. Intuitively, these are queries that pass probes for all depths $q < p$ and are robustly-independent from previous depth-p exploratory queries. We recall the notation n_p for the number of depth-p exploratory queries. These are denoted by $\boldsymbol{y}_1^{(p)}, \dots, \boldsymbol{y}_{n_p}^{(p)}$.

Input: depth P; dimensions d, $\tilde{d}$, $l_1 \ldots, l_P$; number of exploratory queries k; algorithm alg

1 Sample independently $E_1, \ldots, E_P$, uniform $\tilde{d}$-dimensional subspaces of $\mathbb{R}^d$
2 Initialize $n_p \leftarrow 0$ for $p \in [P]$ and set memory of alg to $\mathbf{0}$
3 **while** *alg has not queried a successful point (in $Q_{E_1,\ldots,E_P}$)* **do**
4 Run alg with current memory to obtain query $\boldsymbol{x}$
5 **for** $p \in [P]$ **do**
6 **if** $\boldsymbol{x}$ *is a depth-p exploratory query, i.e., satisfies Eq (9.4) and (9.5)* **then**
7 **if** $n_p < k$ **then**
8 $n_p \leftarrow n_p + 1$
9 $\boldsymbol{y}_{n_p}^{(p)} \leftarrow \boldsymbol{x}$ and sample $V_{n_p}^{(p)}$ a uniform l_p-dimensional subspace of E_p
10 **else**
11 Reset $n_q \leftarrow 1$ for $q \in [p]$
12 $\boldsymbol{y}_1^{(q)} \leftarrow \boldsymbol{x}$ and sample $V_1^{(q)}$ a uniform l_q-dimensional subspace of E_q for $q \in [p]$
13 **end**
14 **if** $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}) \neq$ Success **then return** $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x})$ as response to alg;
15 **else return** $\mathcal{O}_{E_1,\ldots,E_P}(\boldsymbol{x})$ as response to alg;
16 **end**

Procedure 9.3: Adaptive separation oracle for optimization algorithm alg

We are now ready to introduce the complete procedure to construct the adaptive separation oracles, which is formally detailed in Procedure 9.3. The procedure has $P \geq 1$ levels, each of which is associated to a $\tilde{d}$-dimensional subspace $E_1, \ldots, E_P$ that one needs to query perpendicular queries to. Whenever a new depth-p exploratory query is made, we sample a new l_p-dimensional linear subspace $V_{n_p+1}^{(p)}$ uniformly in E_p unless there were already $n_p = k$ such subspaces. In that case, we reset all exploratory queries at depth p as well as all depths $q \leq p$, we pose $n_q = 1$, and sample new subspaces $V_1^{(q)}$. We recall that $l_p = l$ except for $p = P$. For each level we denote by $\boldsymbol{V}^{(p)} := (V_1^{(p)}, \ldots, V_{n_p}^{(p)})$ the list of current linear subspaces at depth p. We next recall the form of the oracle $\mathcal{O}_{\boldsymbol{V}^{(1)},\ldots,\boldsymbol{V}^{(P)}}$ from Eq (9.6). When this oracle returns a vector of the form $\boldsymbol{g}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \delta_p)$ for some $p \in [P]$, we say that $\boldsymbol{x}$ was a depth-p query. For simplicity, we may use the shorthand $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x})$ where $\boldsymbol{V} = (\boldsymbol{V}^{(1)}, \ldots, \boldsymbol{V}^{(P)})$ is the collection of the previous sequences when there is no ambiguity.

Note that $\mathcal{O}_{(E_1),\ldots,(E_P)}$ is a valid separation oracle for $Q_{E_1,\ldots,E_P}$. By abuse of notation, we will simply write it as $\mathcal{O}_{E_1,\ldots,E_P}$. Indeed, we can check that

$$\mathcal{O}_{E_1,\ldots,E_P}(\boldsymbol{x}) = \begin{cases} \boldsymbol{e} & \text{if } \boldsymbol{e}^\top \boldsymbol{x} > -\frac{1}{2} \\ \frac{\mathrm{Proj}_{E_p}(\boldsymbol{x})}{\|\mathrm{Proj}_{E_p}(\boldsymbol{x})\|} & \text{if } \|\mathrm{Proj}_{E_p}(\boldsymbol{x})\| > \delta_p \text{ and } p = \min\left\{q \in [P], \|\mathrm{Proj}_{E_q}(\boldsymbol{x})\| > \delta_q\right\} \\ \mathsf{Success} & \text{otherwise.} \end{cases}$$

We will fall back to this simple separation oracle for $Q_{E_1,\ldots,E_P}$ whenever the oracles from $\mathcal{O}_{\boldsymbol{V}^{(1)},\ldots,\boldsymbol{V}^{(P)}}$ return Success (lines 14-15 of Procedure 9.3).

We first check that Procedure 9.3 indeed corresponds to a valid run for a feasibility problem and specify its corresponding accuracy parameter.

Lemma 9.1. *If $P \geq 2$ and $d \geq 40P$, then with probability at least $1 - e^{-d/40}$ over the randomness of $E_1, \ldots, E_P$, the set $Q_{E_1,\ldots,E_P}$ contains a ball of radius $\delta_1/2$ and all responses of the procedure are valid separating hyperplanes for this feasible set.*

Proof Note that $\mathcal{O}_{\boldsymbol{V}}$ either returns Success or outputs a separating hyperplane for $Q_{E_1,\ldots,E_P}$. When $\mathcal{O}_{\boldsymbol{V}}$ outputs Success, the procedure instead uses an arbitrary valid separation oracle $\mathcal{O}_{E_1,\ldots,E_P}$ (line 15 of Procedure 9.3). We now bound the accuracy parameter ϵ. Because $E_1 \ldots, E_P$ are sampled uniformly randomly, on an event $\mathcal{E}$ of probability one, they are all linearly independent. Hence $Span(E_p, p \in [P])^\perp$ has dimension $d - \tilde{d}P \geq d/2$ and this space is uniformly randomly distributed. In particular, Lemma 9.14 implies that with $\boldsymbol{f} := \mathrm{Proj}_{Span(E_p, p\in[P])^\perp}(\boldsymbol{e})$,

$$\mathbb{P}\left(\|\boldsymbol{f}\| \leq \frac{1}{2} + \frac{1}{40}\right) \leq \mathbb{P}\left(\|\boldsymbol{f}\| \leq \sqrt{\frac{d - \tilde{d}P}{d}\left(1 - \frac{1}{\sqrt{5}}\right)}\right) \leq e^{-(d-\tilde{d}P)/20} \leq e^{-d/40}.$$

Denote by $\mathcal{F}$ the complement of this event in which $\|\boldsymbol{f}\| > \frac{1}{2} + \frac{1}{40}$. Then, under $\mathcal{E} \cap \mathcal{F}$ which has probability at least $1 - e^{-d/72}$,

$$B_d(\mathbf{0}, 1) \cap B_d\left(\boldsymbol{f}, \delta_1 \wedge 1/40\right) \subset Q_{E_1,\ldots,E_P}.$$

Note that $\delta_1 \leq 1/240$. Hence the left-hand side is simply $B_d(\mathbf{0}, 1) \cap B_d(\boldsymbol{f}, \delta_1)$. Now because $\boldsymbol{f} \in B_d(\mathbf{0}, 1)$, this intersection contains a ball of radius $\delta_1/2$. ■

9.4.2 Construction of the feasibility game for all depths

Before trying to prove query lower bounds for memory-constrained algorithms under the feasibility Procedure 9.3, we first define a few concepts. For any $p \in [P]$, we recall that a depth-p *period* as the interval $[t_1, t_2)$ between two consecutive times $t_1 < t_2$ when n_p was reset—we consider that it was also reset at time $t = 1$. Except for time $t = 1$, the reset happens at lines 11-12 of Procedure 9.3. Note that at the beginning of a period, all probing subspaces $V_i^{(p)}$ for $i \in [k]$ are also reset (they will be overwritten during the period). We say that the period $[t_1, \ldots, t_2)$ is *complete* if during this period the algorithm queried k depth-p exploratory queries, or equivalently, if at any time during this period one had $n_p = k$.

The query lower bound proof uses an induction argument on the depth $p \in [P]$. Precisely, we aim to show a lower bound on the number of iterations needed to complete a period for some depth $p \in [P]$. To do so, instead of working directly with Procedure 9.3, we prove lower bounds on the following Depth-p Game 9.4 for $p \in [P]$. Intuitively, it emulates the run of a completed depth-p period from the original feasibility procedure except for the following main points.

1. We allow the learner to have access to some initial memory about the subspaces $E_1, \ldots, E_P$. This intuitively makes it simpler for the player.

Input: game depth $p \in [P]$; number of exploratory queries k; dimensions d, $\tilde{d}$, $l_1, \ldots, l_P$; M-bit memory algorithm alg; number queries T_{max}

1 *Oracle:* Sample independently $E_1, \ldots, E_P$, uniform $\tilde{d}$-dimensional linear subspaces of $\mathbb{R}^d$ and for $i \in [P-p]$ sample k uniform l_{p+i}-dimensional subspaces of E_{p+i}: $V_j^{(p+i)}$ for $j \in [k]$

2 *Player:* Observe $E_1, \ldots, E_P$ and $V_j^{(p+i)}$ for $i \in [P-p]$, $j \in [k]$. Submit to oracle an M-bit message Message, and for all $i \in [P-p]$ submit an integer $n_{p+i} \in [k]$ and vectors $\boldsymbol{y}_j^{(p+i)} \in \mathbb{R}^d$ for $j \in [n_{p+i}]$

3 *Oracle:* Initialize memory of alg to Message. Set $n_{p'} \leftarrow 0$ for $p' \in [p]$ and for $i \in [P-p]$, and n_{p+i} as submitted by player. Reset probing subspaces $V_j^{(p+i)}$ for $i \in [P-p]$ and $j > n_{p+i}$

4 *Oracle:* **for** $t \in [T_{max}]$ **do**

5 Run alg with current memory to get query $\boldsymbol{x}_t$

6 Update exploratory queries $\boldsymbol{y}_i^{(p')}$ and subspaces $V_i^{(p')}$ for $p' \in [P], i \in [n_{p'}]$ as in Procedure 9.3 and **return** $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg

7 **if** n_p *was reset because a deeper period was completed (*$n_p \leftarrow 1$ *in line 12 of Procedure 9.3)* **then** player loses, **end** game;

8 **if** $n_p = k$ **then** player wins. **end** game;

9 **end**

10 Player loses

Game 9.4: Depth-p Feasibility Game

2. The learner can also submit some vectors $\boldsymbol{y}_j^{(p+i)}$ for $i \in [P-p]$ that emulate exploratory queries for deeper depths than p.

3. The objective is not to find a point in the feasible set $Q_{E_1,\ldots,E_P}$ anymore, but simply to complete the depth-p period, that is, make k depth-p exploratory queries. Note that the player loses if a deeper period was completed (line 7 of Game 9.4). Hence, during a winning run, no depth-p' periods with $p' > p$ were completed.

4. The player has a maximum number of calls to the separation oracle available T_{max}.

Note that the role of the player here is only to submit the message Message and exploratory queries for depths $p' > p$ to the oracle, in addition to providing a M-bit memory algorithm alg. In the second part of Game 9.4, the oracle directly emulates a run of a depth-p period of Procedure 9.3 (without needing input from the player). We will also use the term depth-q period for $q \in [p]$ for this game. One of the interests of the third point is to make the problem symmetric in terms of the objectives at depth $p \in [P]$, which will help implement our recursive query lower bound argument.

9.4.3 Properties of the probing subspaces

The probing subspaces $V_i^{(q)}$ at any level $q \in [P]$ are designed to "probe" for the query $\boldsymbol{x}$ being close to the perpendicular space to E_q. Precisely, the first step of the proof is to show that if a query $\boldsymbol{x}_t$ passed probes at level q, then with high probability it satisfied $\|\mathrm{Proj}_{E_p}(\boldsymbol{x}_t)\| \le \eta_p$. To prove this formally, we introduce the Probing Game 9.1. It mimics Procedure 9.3 but focuses on exploratory queries at a single layer. We recall its definition here for the sake of exposition.

Input: dimension d, response dimension l, number of exploratory queries k, objective $\rho > 0$

Oracle: Sample independent uniform l-dimensional subspaces $V_1, \dots, V_k$ in $\mathbb{R}^d$.
for $i \in [k]$ **do**
  Player: Based on responses $V_j, j < i$, submit query $\boldsymbol{y}_i \in \mathbb{R}^d$
  Oracle: **return** V_i to the player
end
Player wins if for any $i \in [k]$ there exists a vector $\boldsymbol{z} \in Span(\boldsymbol{y}_j, j \in [i])$ such that

$$\|\mathrm{Proj}_{Span(V_j, j\in[i])}(\boldsymbol{z})\| \le \rho \quad \text{and} \quad \|\boldsymbol{z}\| = 1.$$

Game 9.1: Probing Game

Our goal is to give a bound on the probability of success of any strategy for Game 9.1. To do so, we need to prove the Theorem 9.3 which essentially bounds the smallest singular value of random matrices for which the upper triangle components are all i.i.d. standard normal random variables. We start by proving the result in the case when the lower triangle is identically zero, which may be of independent interest.

Corollary 9.1. *Let $C \ge 2$ be an integer and $m = Cn$. Let $\boldsymbol{M} \in \mathbb{R}^{n\times m}$ be the random matrix such that all coordinates $M_{i,j}$ in the upper-triangle $j > (i-1)C$ are together i.i.d. Gaussians $\mathcal{N}(0,1)$ and the lower triangle is zero, that is $M_{i,j} = 0$ for $j \le (i-1)C$. Then for any $\alpha \in (0,1]$, if $C \ge C_\alpha \ln n$, we have*

$$\mathbb{P}\left(\sigma_1(\boldsymbol{M}) < \frac{1}{6}\sqrt{\frac{C}{n^\alpha}}\right) \le 3e^{-C/16}.$$

Here $C_\alpha = (C_2/\alpha)^{\ln 2/\alpha}$ for some universal constant $C_2 \ge 8$.

Proof We use an ϵ-net argument to prove this result. However, we will need to construct the argument for various scales of ϵ because of the non-homogeneity of the triangular matrix $\boldsymbol{M}$. First, we can upper bound the maximum singular value of $\boldsymbol{M}$ directly as follows (see e.g. [Ver20, Theorem 4.4.5] or [Tao23, Exercise 2.3.3]),

$$\mathbb{P}(\|\boldsymbol{M}\|_{op} > C_1\sqrt{Cn}) \le e^{-Cn}$$

for some universal constant $C_1 \ge 1$. For convenience, let us write

$$\boldsymbol{M}(i) := (M_{u,v})_{u\in[i,n], v\in[1+C(i-1),Cn]}, \quad i \in [n].$$

Applying the above result implies that

$$\mathbb{P}(\|\boldsymbol{M}(i)\|_{op} > C_1\sqrt{C(n-i+1)}) \leq e^{-C(n-i+1)}, \quad i \in [n].$$

In particular, the event

$$\mathcal{E} = \bigcap_{i\in[n]} \left\{ \|\boldsymbol{M}(i)\|_{op} \leq C_1\sqrt{C(n-i+1)} \right\} \tag{9.10}$$

satisfies $\mathbb{P}(\mathcal{E}^c) \leq \sum_{i\in[n]} e^{-C(n-i+1)} \leq 2e^{-C}$ because $C \geq \ln(2)$. Next, let $\epsilon = 1/(6nC_1)$ and fix a constant $\alpha \in (0,1)$. For any $i \in [n]$, we construct an ϵ-net Σ_i of all unit vectors which have non-zero coordinates in $[i,n]$, that is $S_{d-1} \cap \{\boldsymbol{x} : \forall j < i, x_j = 0\}$. For any $\boldsymbol{x} \in S_{d-1}$ note that

$$\|\boldsymbol{M}^\top \boldsymbol{x}\|^2 \sim \sum_{i=1}^n Y_i \cdot \|(x_j)_{j\in[i]}\|^2,$$

where $Y_1, \dots, Y_n \overset{i.i.d.}{\sim} \chi^2(C)$ are i.i.d. chi-squared random variables. Hence, for and $i \in [n]$, together with Lemma 9.13 this shows that

$$\mathbb{P}\left(\|\boldsymbol{M}^\top \boldsymbol{x}\| < \frac{1}{2}\sqrt{C(n-i+1)\sum_{j\in[i]} x_j^2} \right) \leq \mathbb{P}\left(\sum_{j=i}^n Y_j < \frac{C(n-i+1)}{4} \right) \leq e^{-C(n-i+1)/8}.$$

In the last inequality, we used the fact that $\sum_{j=i}^n Y_i \sim \chi^2(C(n-i+1))$ hence has the distribution of the squared norm of a Gaussian $\mathcal{N}(0, \boldsymbol{I}_{n-i+1})$. Next, because Σ_i is an ϵ-net of a sphere of dimension $n-i+1$ (restricted to the last $n-i+1$ coordinates) there is a universal constant $C_2 \geq 1$ such that $|\Sigma^{(i)}| \leq (C_2/\epsilon)^{n-i+1}$ (e.g. see [Tao23, Lemma 2.3.4]. For any $i \in [n]$, we write $l(i) = 1 \vee (n - L_\alpha(n-i+1)+1)$ for a constant integer $L_\alpha \geq 2(C_1/\alpha)^{\ln 2/\alpha}$ to be fixed later. Note that by construction we always have $n - l(i) + 1 \leq L_\alpha(n-i+1)$. Finally, we define the event

$$\mathcal{F} = \bigcap_{i\in[n]} \bigcap_{\boldsymbol{x}\in\Sigma_{l(i)}} \left\{ \|\boldsymbol{M}^\top \boldsymbol{x}\| \geq \frac{1}{2}\sqrt{C(n-i+1)\sum_{j\in[i]} x_j^2} \right\}.$$

Provided that $e^{C/16} \geq 2(C_2/\epsilon)^{L_\alpha}$ the union bound implies

$$\mathbb{P}(\mathcal{F}^c) \leq \sum_{i=1}^n |\Sigma_{l(i)}| e^{-C(n-i+1)/8} \leq \sum_{i=1}^n \left(\frac{C_2^{L_\alpha}}{\epsilon^{L_\alpha} e^{C/8}} \right)^{n-i+1} \leq e^{-C/16}.$$

We note that the equation $e^{C/16} \geq 2(C_2/\epsilon)^{L_\alpha}$ is equivalent to $C/16 \geq L_\alpha \ln(6C_1C_2\sqrt{n}) + \ln 2$, which is satisfied whenever $C \geq C_3 L_\alpha \ln n$ for some universal constant $C_3 \geq 16$. We suppose that this is the case from now on and that the event $\mathcal{E} \cap \mathcal{F}$ is satisfied.

We now construct a sequence of growing indices as follows. For all $k \leq \lfloor \log_2(n) \rfloor - 1 := k_0$, we let $i_k := n - 2^k + 1$ for $k < k_0$. For any $\boldsymbol{x} \in S_{d-1}$, we define

$$k(\boldsymbol{x}) := \arg\max_{k\in\{0,\dots,k_0\}} e^{\alpha k} f(k;\boldsymbol{x}) \quad \text{where} \quad f(k;\boldsymbol{x}) := (n - i_k + 1)\sum_{j\in[i_k]} x_j^2.$$

Note that because $i_0 = n$, the inner maximization problem has value at least $\sum_{j\in[n]} x_j^2 = 1$. Also, for any $k \geq k(\boldsymbol{x})$, we have

$$f(k;\boldsymbol{x}) \leq e^{-\alpha(k-k(\boldsymbol{x}))} f(k(\boldsymbol{x});\boldsymbol{x}). \tag{9.11}$$

We will also use the shortcut $l(\boldsymbol{x}) = l(i_{k(\boldsymbol{x})})$. Let $\hat{\boldsymbol{x}}$ be the nearest neighbor of $\boldsymbol{x}$ in $\Sigma_{l(\boldsymbol{x})}$, that is the nearest neighbor of the vector $\boldsymbol{y} = (x_j \mathbb{1}_{j\geq l(\boldsymbol{x})})_{j\in[n]}$. By construction of the ϵ-nets we have $\|\boldsymbol{y} - \hat{\boldsymbol{x}}\| \leq \epsilon$. Now observe that $\boldsymbol{x} - \boldsymbol{y}$ only has non-zero values for the first $l(\boldsymbol{x}) - 1$ coordinates. We let $r(\boldsymbol{x})$ be the index such that $i_{r(\boldsymbol{x})+1} < l(\boldsymbol{x}) - 1 \leq i_{r(\boldsymbol{x})}$ (with the convention $i_{k_0+1} = 0$). We decompose $\boldsymbol{x} - \boldsymbol{y}$ as the linear sum of vectors for $r \in [r(\boldsymbol{x}), k_0]$ such that the r-th vector only has non-zero values for coordinates in $(i_{r+1}, i_r]$. Then, using the triangular inequality

$$\|\boldsymbol{M}^\top \boldsymbol{x}\| \geq \|\boldsymbol{M}^\top \hat{\boldsymbol{x}}\| - \|\boldsymbol{M}^\top(\boldsymbol{y} - \hat{\boldsymbol{x}})\| - \sum_{r=r(\boldsymbol{x})}^{k_0} \|\boldsymbol{M}^\top (x_j \mathbb{1}_{i_{r+1}<j\leq i_r \wedge (l(\boldsymbol{x})-1)})_{j\in[n]}\| \tag{9.12}$$

$$\geq \|\boldsymbol{M}^\top \hat{\boldsymbol{x}}\| - \epsilon \|\boldsymbol{M}(l(\boldsymbol{x}))^\top\|_{op} - \sum_{r=r(\boldsymbol{x})}^{k_0} \|\boldsymbol{M}(i_r)\|_{op} \sqrt{\sum_{j\in[i_r]} x_j^2} \tag{9.13}$$

$$\geq \|\boldsymbol{M}^\top \hat{\boldsymbol{x}}\| - C_1 \epsilon \sqrt{Cn} - C_1 \sum_{r=r(\boldsymbol{x})}^{k_0} \sqrt{C \cdot f(r;\boldsymbol{x})}. \tag{9.14}$$

In the last inequality we used Eq (9.10). We start by treating the second term containing ϵ. We recall that the inner maximization problem defining $k(\boldsymbol{x})$ has value at least 1, so that

$$f(k(\boldsymbol{x});\boldsymbol{x}) \geq e^{-\alpha k(\boldsymbol{x})} \geq e^{-\alpha k_0} \geq n^{-\alpha}. \tag{9.15}$$

As a result, recalling that $\alpha \in (0, 1]$, we have

$$C_1 \epsilon \sqrt{Cn} \leq \frac{1}{6}\sqrt{\frac{C}{n}} \leq \frac{\sqrt{C \cdot f(k(\boldsymbol{x});\boldsymbol{x})}}{6}. \tag{9.16}$$

Next, by Eq (9.11) we have

$$\sum_{r=r(\boldsymbol{x})}^{k_0} \sqrt{f(r;\boldsymbol{x})} \leq \sqrt{f(k(\boldsymbol{x});\boldsymbol{x})} \sum_{r=r(\boldsymbol{x})}^{k_0} e^{-\alpha(r-k(\boldsymbol{x}))/2} \leq \frac{e^{-\alpha(r(\boldsymbol{x})-k(\boldsymbol{x}))/2}}{1 - e^{-\alpha/2}} \sqrt{f(k(\boldsymbol{x});\boldsymbol{x})}$$
$$\leq \frac{4}{\alpha} e^{-\alpha(r(\boldsymbol{x})-k(\boldsymbol{x}))/2} \sqrt{f(k(\boldsymbol{x});\boldsymbol{x})}.$$

In the last inequality, we used the fact that $\alpha \in (0, 1)$ and that $e^{-x} \leq 1 - x/2$ for all $x \in [0, 1]$. Next, recall that $i_{r(\boldsymbol{x})+1} < l(\boldsymbol{x}) - 1$. Hence, either $l(\boldsymbol{x}) = 1$ in which case the sum above is empty, or we have

$$2^{r(\boldsymbol{x})+1} = n - i_{r(\boldsymbol{x})+1} + 1 \geq n - l(\boldsymbol{x}) + 1 = L_\alpha(n - i_{k(\boldsymbol{x})} + 1) = L_\alpha 2^{k(\boldsymbol{x})}.$$

As a result, $r(\boldsymbol{x}) - k(\boldsymbol{x}) \geq \log_2 L_\alpha - 1$. We now select L_α to be the minimum integer for which $\log_2 L_\alpha - 1 \geq \frac{2\ln(24C_1/\alpha)}{\alpha}$. In turn, this implies

$$C_1 \sum_{r=r(\boldsymbol{x})}^{k_0} \sqrt{C \cdot f(r;\boldsymbol{x})} \leq \frac{\sqrt{C \cdot f(k(\boldsymbol{x});\boldsymbol{x})}}{6}. \tag{9.17}$$

Putting together Eq (9.14), (9.16), (9.17) we obtained

$$\|\boldsymbol{M}^\top \boldsymbol{x}\| \geq \|\boldsymbol{M}^\top \hat{\boldsymbol{x}}\| - \frac{1}{3}\sqrt{C \cdot f(k(\boldsymbol{x});\boldsymbol{x})} \geq \frac{1}{6}\sqrt{C \cdot f(k(\boldsymbol{x});\boldsymbol{x})} \geq \frac{1}{6}\sqrt{\frac{C}{n^\alpha}}.$$

where in the second inequality, we used the event $\mathcal{F}$ and the last inequality used Eq (9.15). In summary, we showed that if $C \geq C_2 L_\alpha \ln n$, then

$$\mathbb{P}\left(\sigma_1(\boldsymbol{M}) \geq \frac{1}{6}\sqrt{\frac{C}{n^\alpha}}\right) \geq \mathbb{P}(\mathcal{E} \cap \mathcal{F}) \geq 1 - 3e^{-C/16}.$$

This ends the proof of the result. ■

We are now ready to prove Theorem 9.3 which generalizes Corollary 9.1 to some cases when the lower triangle can be dependent on the upper triangle.

Proof of Theorem 9.3 The main part of the proof reduces the problem to the case when the lower-triangle is identically zero. To do so, let $\boldsymbol{x} \sim \mathcal{U}(S_{n-1})$ be a random unit vector independent of $\boldsymbol{M}$. We aim to lower bound $\|\boldsymbol{M}^\top \boldsymbol{x}\|$. Let us define some notations for sub-matrices and subvectors of $\boldsymbol{M}$ as follows for any $i \in [n]$,

$$\begin{aligned}
\boldsymbol{M}^{(i)} &:= (M_{u,v})_{u\in[i],v\in[Ci]} \\
\boldsymbol{N}^{(i)} &:= (M_{u,v})_{u\in[i],v\in[C(i-1)]} \\
\boldsymbol{a}^{(i)} &:= (M_{i,v})_{v\in[C(i-1)]}, \\
\boldsymbol{A}^{(i)} &:= (M_{u,v})_{u\in[i],v\in[C(i-1)+1,C_i]}.
\end{aligned}$$

For a visual representation of these submatrices, we have the following nested construction

$$\boldsymbol{M}^{(i)} = \left[\begin{array}{c|c} \boldsymbol{M}^{(i-1)} & \boldsymbol{A}^{(i)} \\ \hline {\boldsymbol{a}^{(i)}}^\top & \end{array}\right] = [\boldsymbol{N}^{(i)}, \boldsymbol{A}^{(i)}]\,, \quad i \in [n]$$

and $\boldsymbol{M}^{(n)} = \boldsymbol{M}$. For now, fix $i \in [2, n]$. We consider any realization of the matrix $\boldsymbol{N}^{(i)}$, that is, of the matrix $M^{(i-1)}$ and the vector $\boldsymbol{a}^{(i)}$. Then,

$${\boldsymbol{N}^{(i)}}^\top \boldsymbol{N}^{(i)} = {\boldsymbol{M}^{(i-1)}}^\top \boldsymbol{M}^{(i-1)} + \boldsymbol{a}^{(i)}{\boldsymbol{a}^{(i)}}^\top \succeq {\boldsymbol{M}^{(i-1)}}^\top \boldsymbol{M}^{(i-1)}. \tag{9.18}$$

Next, let $0 \leq \sigma_1 \leq \ldots \leq \sigma_i$ be the singular values of the matrix $\mathbb{N}^{(i)}$. We also define

$$\tilde{\boldsymbol{N}}^{(i)} := [{\boldsymbol{M}^{(i-1)}}^\top, \boldsymbol{0}_{C(i-1),1}]^\top,$$

which is exactly the matrix $\boldsymbol{N}^{(i)}$ if we had $\boldsymbol{a}^{(i)} = \boldsymbol{0}$, and let $0 \leq \tilde{\sigma}_1 \leq \ldots \leq \tilde{\sigma}_i$ be the singular values of $\tilde{\boldsymbol{N}}^{(i)}$. Then, Eq (9.18) implies that for all $j \in [i]$, one has $\tilde{\sigma}_j \geq \sigma_j$. After constructing the singular value decomposition of the two matrices $\boldsymbol{N}^{(i)}$ and $\tilde{\boldsymbol{N}}^{(i)}$, this shows that there exists an orthogonal matrix $\boldsymbol{U}^{(i)} \in \mathcal{O}_i$ such that

$$\|\boldsymbol{N}^{(i)\top}\boldsymbol{x}^{(i)}\| \geq \|\tilde{\boldsymbol{N}}^{(i)\top}(\boldsymbol{U}^{(i)}\boldsymbol{x}^{(i)})\|, \quad \forall \boldsymbol{x}^{(i)} \in \mathbb{R}^i.$$

Finally, defining the matrix

$$\tilde{\boldsymbol{M}}^{(i)} = [\tilde{\boldsymbol{N}}^{(i)}, \boldsymbol{U}^{(i)}\boldsymbol{A}^{(i)}] = \left|\begin{array}{c|c} \boldsymbol{M}^{(i-1)} & \boldsymbol{U}^{(i)}\boldsymbol{A}^{(i)} \\ \hline 0 & \end{array}\right|, \tag{9.19}$$

we obtained that

$$\|\boldsymbol{M}^{(i)\top}\boldsymbol{x}^{(i)}\| \geq \|\tilde{\boldsymbol{M}}^{(i)\top}(\boldsymbol{U}^{(i)}\boldsymbol{x}^{(i)})\|, \quad \forall \boldsymbol{x}^{(i)} \in \mathbb{R}^i.$$

We now make a few simple remarks. First, the uniform distribution on S_{i-1} is invariant by the rotation $\boldsymbol{U}^{(i)}$, hence $\boldsymbol{x}^{(i)} \sim \mathcal{U}(S_{i-1})$ implies $\boldsymbol{U}^{(i)}\boldsymbol{x}^{(i)} \sim \mathcal{U}(S_{i-1})$. Next, by isometry of Gaussian vectors, conditionally on $\boldsymbol{N}^{(i)}$, the matrix $\tilde{\boldsymbol{A}}^{(i)} := \boldsymbol{U}^{(i)}\boldsymbol{A}^{(i)}$ is still distributed exactly as a matrix with i.i.d. $\mathcal{N}(0,1)$ entries. In turn, $\|\tilde{\boldsymbol{A}}^{(i)\top}(\boldsymbol{U}^{(i)}\boldsymbol{x}^{(i)})\|^2$ is still distributed as a chi-squares $\chi^2(C)$ independent from $\boldsymbol{N}^{(i)}$, just as for $\|\boldsymbol{A}^{(i)\top}\boldsymbol{x}^{(i))}\|^2$. As a summary of the past arguments, using the notation $\succeq_{st}$ for stochastic dominance we obtained for $\boldsymbol{x}^{(i)} \sim \mathcal{U}(S_{i-1})$ sampled independently from $\boldsymbol{M}$,

$$\|\boldsymbol{M}^{(i)\top}\boldsymbol{x}^{(i)}\|^2 \succeq_{st} \|\tilde{\boldsymbol{M}}^{(i)\top}\boldsymbol{x}^{(i)}\|^2 = \|\boldsymbol{M}^{(i-1)\top}(x_j^{(i)})_{j\in[i-1]}\|^2 + \|\tilde{\boldsymbol{A}}^{(i)\top}\boldsymbol{x}^{(i)}\|^2.$$

As a result,

$$\|\boldsymbol{M}^{(i)\top}\boldsymbol{x}^{(i)}\|^2 \succeq_{st} (1-(x_i^{(i)})^2)\|\boldsymbol{M}^{(i-1)^\top}\boldsymbol{x}^{(i-1)}\|^2 + Y_i, \tag{9.20}$$

where $\boldsymbol{x}^{(i-1)} \sim \mathcal{U}(S_{i-2})$ (can be resampled independently from $\boldsymbol{x}^{(i)}$ if wanted) and $Y_i \sim \chi^2(C)$ is independent from $\boldsymbol{M}^{(i-1)}$, $x_i^{(i)}$ and $\boldsymbol{x}^{(i-1)}$.

Using Eq (9.20) recursively constructs a sequence of random independent vectors $\boldsymbol{x}^{(i)} \sim \mathcal{U}(S_{i-1})$ for $i \in [n]$, as well as an independent sequence of i.i.d. $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \chi^2(C)$ such that for $\boldsymbol{x} \sim \mathcal{U}(S_{n-1})$ sampled independently from $\boldsymbol{M}$ and $Y_1, \ldots, Y_n$,

$$\|\boldsymbol{M}^\top\boldsymbol{x}\|^2 \succeq_{st} \sum_{i=1}^n Y_i \prod_{j=i+1}^n (1-(x_j^{(j)})^2) \sim \sum_{i=1}^n Y_i \cdot \|(x_j)_{j\in[i]}\|^2. \tag{9.21}$$

The last inequality holds because of the observation that if $\boldsymbol{x} \sim \mathcal{U}(S_{i-1})$, then the first $i-1$ coordinates of $\boldsymbol{x}$ are distributed as a uniform vector $\mathcal{U}(S_{i-2})$ rescaled by $\sqrt{1-x_i^2}$. Next, we construct the matrix $\boldsymbol{M}^0 \in \mathbb{R}^{n\times Cn}$ that corresponds exactly to $\boldsymbol{M}$ had the vectors $\boldsymbol{a}^{(i)}$ been all identically zero: all coordinates $M^0_{i,j}$ for $j > (i-1)C$ are i.i.d. $\mathcal{N}(0,1)$ and for $j < (i-1)C$ we have $M^0_{i,j} = 0$. We can easily check that

$$\|\boldsymbol{M}^{0^\top}\boldsymbol{x}\|^2 \sim \sum_{i=1}^n Y_i \cdot \|(x_j)_{j\in[i]}\|^2. \tag{9.22}$$

Combining this equation together with Eq (9.21) shows that

$$\|\boldsymbol{M}^\top \boldsymbol{x}\| \succeq_{st} \|{\boldsymbol{M}^0}^\top \boldsymbol{x}\|, \quad \boldsymbol{x} \sim \mathcal{U}(S_{n-1}).$$

Hence, intuitively the worst case to lower bound the singular values of $\boldsymbol{M}$ corresponds exactly to the case when all the vectors $\boldsymbol{a}^{(i)}$ were identically zero. While the previous equation concisely expresses this idea, we need a slightly stronger statement that also characterizes the form of the coupling between $(\boldsymbol{M}, \boldsymbol{x})$ and $(\boldsymbol{M}^0, \tilde{\boldsymbol{x}})$ such that almost surely, $\|\boldsymbol{M}^\top \boldsymbol{x}\| \geq \|{\boldsymbol{M}^0}^\top \tilde{\boldsymbol{x}}\|$. Going back to the construction of these couplings, we note that $\tilde{\boldsymbol{x}}$ is obtained from $\boldsymbol{x}$ by applying a sequence of rotations to some of its components—the matrices $\boldsymbol{U}^{(i)}$—and these rotations only depend on $\boldsymbol{M}$. Similarly, the matrix $\boldsymbol{M}^0$ is coupled to $\boldsymbol{M}$ but is independent from $\boldsymbol{x}$ (see Eq (9.19)). The last remark is crucial because it shows that having fixed a realization for $\boldsymbol{M}$ and $\boldsymbol{M}^0$, for $\boldsymbol{x} \sim \mathcal{U}(S_{d-1})$ sampled independently from $\boldsymbol{M}$ we have

$$\|\boldsymbol{M}^\top \boldsymbol{x}\| \geq \|{\boldsymbol{M}^0}^\top \tilde{\boldsymbol{x}}\| \geq \sigma_1(\boldsymbol{M}^0).$$

As a result, the above equation holds for almost all $\boldsymbol{x} \in S_{d-1}$, which is sufficient to prove that almost surely, $\sigma_1(\boldsymbol{M}) \geq \sigma_1(\boldsymbol{M}^0)$. Together with the lower bound on $\sigma_1(\boldsymbol{M}^0)$ from Corollary 9.1 we obtain the desired result. ■

Having these random matrices results at hand, we can now give query lower bounds on the Probing Game 9.1.

Theorem 9.5. *Suppose that $4kl \leq d$ and $l \geq C_\alpha \ln k$ for a fixed $\alpha \in (0, 1]$, where C_α is as defined in Theorem 9.3. Let $\rho \leq \frac{1}{12}\sqrt{\frac{l}{dk^\alpha}}$. Then, no algorithm wins at the Probing Game 9.1 with probability at least $4de^{-l/16}$.*

Proof of Theorem 9.5 We start with defining some notations. We observe that for any $i \in [k]$, sampling V_i uniformly within l-dimensional subspaces of $\mathbb{R}^d$ is stochastically equivalent to sampling some Gaussian vectors $\boldsymbol{v}_i^{(1)}, \ldots, \boldsymbol{v}_i^{(l)} \overset{i.i.d.}{\sim} \mathcal{N}(0, \boldsymbol{I}_d)$ and constructing the subspace $Span(\boldsymbol{v}_i^{(r)}, r \in [l])$. Without loss of generality, we can therefore suppose that the probing subspaces were sampled in this way. We then define a matrix $\boldsymbol{\Pi}$ summarizing all probing subspaces as follows:

$$\boldsymbol{\Pi} := [\boldsymbol{v}_1^{(1)}, \ldots, \boldsymbol{v}_1^{(l)}, \boldsymbol{v}_2^{(1)}, \ldots, \boldsymbol{v}_2^{(l)}, \ldots, \boldsymbol{v}_k^{(1)}, \ldots, \boldsymbol{v}_k^{(l)}].$$

Next, we let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ be the sequence resulting from doing the Gram-Schmidt decomposition from $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$, that is

$$\boldsymbol{x}_i = \begin{cases} \frac{\mathrm{Proj}_{Span(\boldsymbol{y}_j, j<i)^\perp}(\boldsymbol{y}_i)}{\|\mathrm{Proj}_{Span(\boldsymbol{y}_j, j<i)^\perp}(\boldsymbol{y}_i)\|} & \text{if } \boldsymbol{y}_i \notin Span(\boldsymbol{y}_j, j < i) \\ \boldsymbol{0} & \text{otherwise.} \end{cases}$$

Note that it is always advantageous for the player to submit a vector $\boldsymbol{y}_i \notin Span(\boldsymbol{y}_j, j < i)$ so that the space $Span(\boldsymbol{y}_j, j \in [i])$ is as large as possible. Without loss of generality, we will

therefore suppose that $\boldsymbol{y}_i \notin Span(\boldsymbol{y}_j, j < i)$ for all $i \in [k]$. In particular, $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ form an orthonormal sequence. Last, we construct the Gram matrix $\boldsymbol{M} \in \mathbb{R}^{k \times lk}$ as

$$\boldsymbol{M} := [\boldsymbol{x}_1, \dots, \boldsymbol{x}_k]^\top \Pi.$$

The next part of this proof is to lower bound the smallest singular value of $\boldsymbol{M}$. To do so, we show that it satisfies the assumptions necessary for Theorem 9.3.

For $i \in [k]$, conditionally on all $\boldsymbol{x}_j$ for $j \in [i]$ as well as all $\boldsymbol{v}_j^{(r)}$ for $j < i$ and $r \in [l]$, the vectors $\boldsymbol{v}_i^{(r)}$ for $r \in [l]$ are still i.i.d. Gaussian vectors $\mathcal{N}(0, \boldsymbol{I}_d)$. Now by construction, $\boldsymbol{x}_1, \dots, \boldsymbol{x}_i$ form an orthonormal sequence. Hence, by isometry of the Gaussian distribution $\mathcal{N}(0, \boldsymbol{I}_d)$, the random variables $(\boldsymbol{x}_j^\top \boldsymbol{v}_i^{(r)})_{j \in [i], r \in [l]}$ are together i.i.d. and distributed as normal random variables $\mathcal{N}(0, 1)$. Because this holds for all $i \in [k]$, this proves that the upper-triangular components of $\boldsymbol{M}$ are all i.i.d. normal $\mathcal{N}(0, 1)$. Next, for any $i \in [k]$, recall that the vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_i$ are independent from all the vectors $\boldsymbol{v}_j^{(r)}$ for $j \geq i$ and $r \in [l]$. In particular, this shows that all components $M_{a,b} = \boldsymbol{x}_a^\top \boldsymbol{v}_b^{(r)}$ for $a \leq i$, $b < i$ and $r \in [l]$ are all independent from the normal variables $M_{a,b} = \boldsymbol{x}_a^\top \boldsymbol{v}_b^{(r)}$ for $a \in [n]$, $b \geq a$ and $b \geq i$ and $r \in [l]$. This exactly shows that $\boldsymbol{M}$ satisfies all properties for Theorem 9.3. In particular, letting $\boldsymbol{M}^{(i)} = (M_{u,v})_{u \in [i], v \in [Ci]}$ for all $i \in [k]$, this also proves that $\boldsymbol{M}^{(i)}$ satisfies the required conditions.

Now fix $\alpha \in (0, 1]$ and suppose that $l \geq C_\alpha \ln k$. In particular, without loss of generality $l \geq 4$. Theorem 9.3 shows that the event

$$\mathcal{E} := \bigcap_{i \in [k]} \left\{ \sigma_1(\boldsymbol{M}^{(i)}) \geq \frac{1}{6} \sqrt{\frac{l}{i^\alpha}} \right\}$$

has probability at least $1 - 3ke^{-l/16}$. In the last part of the proof, we show how to obtain upper bounds on the probability of success of the player for Game 9.1 given these singular values lower bounds. Recall that the vectors $\boldsymbol{v}_i^{(r)}$ for $i \in [k]$ and $r \in [l]$ are all i.i.d. Gaussians $\mathcal{N}(0, \boldsymbol{I}_d)$. Hence, letting $\boldsymbol{\Pi}^{(i)}$ be the matrix that contains the first li columns of $\boldsymbol{\Pi}$ (that is, vectors $\boldsymbol{v}_j^{(r)}$ for $r \in [l]$ and $j \in [i]$), by Theorem 9.14 we have

$$\mathbb{P}\left(\|\boldsymbol{\Pi}^{(i)}\|_{op} \geq \sqrt{d} \left(3/2 + \sqrt{\frac{li}{d}} \right) \right) \leq e^{-d/8}, \quad i \in [k].$$

We note that for any $i \in [k]$, one has $li/d \leq 1/4$. We then define the event

$$\mathcal{F} := \left\{ \|\boldsymbol{\Pi}^{(i)}\|_{op} < 2\sqrt{d}, i \in [k] \right\},$$

which has probability at least $1 - ke^{-d/8}$. On the event $\mathcal{E} \cap \mathcal{F}$, for any $i \in [k]$ and $\boldsymbol{z} \in Span(\boldsymbol{y}_j, j \in [i])$, writing $\boldsymbol{z} = \sum_{j \in [i]} \lambda_j \boldsymbol{x}_j$ we have $\|\boldsymbol{z}\| = \|\boldsymbol{\lambda}\|$ since the sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ is orthonormal. As a result,

$$\|\mathrm{Proj}_{Span(V_j, j \in [i])}(\boldsymbol{z})\| \geq \frac{\|{\boldsymbol{\Pi}^{(i)}}^\top \boldsymbol{z}\|}{\|\boldsymbol{\Pi}^{(i)}\|_{op}} = \frac{\|(\boldsymbol{M}^{(i)})^\top \boldsymbol{\lambda}\|}{\|\boldsymbol{\Pi}^{(i)}\|_{op}} > \frac{\sigma_1(\boldsymbol{M}^{(i)})}{2\sqrt{d}} \|\boldsymbol{z}\| \geq \frac{1}{12} \sqrt{\frac{l}{dk^\alpha}} \|\boldsymbol{z}\|.$$

That is, under $\mathcal{E} \cap \mathcal{F}$ the player does not win for the parameter $\rho = \frac{1}{12}\sqrt{\frac{l}{dk^\alpha}}$. Last, by the union bound, $\mathbb{P}(\mathcal{E} \cap \mathcal{F}) \geq 1 - 3ke^{-l/16} - ke^{-d/8} \geq 1 - 4de^{-l/16}$. This ends the proof. ■

We next define the notion of *proper* period for which the probing subspaces were a good proxy for the projection onto E_q.

Definition 9.3 (Proper periods)**.** *Let $q \in [P]$. We say that a depth-q period starting at time t_1 is* proper *when for any time $t \geq t_1$ during this period, if $\boldsymbol{x}_t$ is a depth-p' query for $p' > q$, then*

$$\|Proj_{E_q}(\boldsymbol{x}_t)\| \leq \eta_q.$$

This definition can apply equally to periods from the Procedure 9.3 or the Depth-p Game 9.4 (provided $q \leq p$). By proving a reduction from running a given period to the Probing Game 9.1, we can show that Theorem 9.5 implies most periods are proper.

Lemma 9.2. *Let $q \in [P]$ ($q \in [p]$ for Depth-p Game 9.4). Suppose that $4l_qk \leq \tilde{d}$ and that $l_q \geq C_\alpha \ln k$ for some fixed constant $\alpha \in (0, 1]$. For any index $j \geq 1$, define the event $\mathcal{E}_q(j) = \{j$ depth-q periods were started$\}$. If $\mathbb{P}(\mathcal{E}_q(j)) > 0$, then,*

$$\mathbb{P}\left(j\text{-th depth-}q \text{ period is proper} \mid \mathcal{E}_q(j)\right) \geq 1 - 4de^{-l_q/16}.$$

Proof We prove the result in the context of Game 9.4 which is the only part that will be needed for the rest of this chapter. The exact same arguments will yield the desired result for Procedure 9.3. We fix a strategy for Game 9.4, a period depth $q \in [p]$, and a period index J such that $\mathbb{P}(\mathcal{E}_q(J)) > 0$. Using this strategy, we construct a learning algorithm for the Probing Game 9.1 with dimension $\tilde{d}$, response dimension l_q, k exploratory queries.

This player strategy is detailed in Algorithm 9.5. It works by simulating a run of Game 9.4, sampling quantities similarly as what the oracle in that game would sample for $E_1, \ldots, E_P$ and the probing subspaces. More precisely, we proceed conditionally on a run of the Game 9.4 starting J depth-q periods. For instance, this can be done by simulating the game until this event happens (Part 1 of Algorithm 9.5). The algorithm then continues the run of the J-th depth-q period using the probing subspaces $V_1, \ldots, V_k$ provided by the oracle of Game 9.1 (Part 2 of Algorithm 9.5). These oracle subspaces and depth-q probing subspaces for Game 9.4 live in different spaces: in $E_q \subset \mathbb{R}^d$ for $V_i^{(q)}$ and $\mathbb{R}^{\tilde{d}}$ for V_i. Hence, we use an isometric mapping $R : \mathbb{R}^d \to \mathbb{R}^d$ whose image of E_q is $\mathbb{R}^{\tilde{d}} \otimes \{0\}^{d-\tilde{d}}$. Letting $\tilde{R} = \pi_{\tilde{d}} \circ R$ where $\pi_{\tilde{d}}$ is the projection onto the first $\tilde{d}$ coordinates, we map any vector $\boldsymbol{x} \in E_q$ to the vector $\tilde{R}(\boldsymbol{x})$ in $\mathbb{R}^{\tilde{d}}$. The natural inverse mapping $\tilde{R}^{<-1>}$ such that $\tilde{R}^{<-1>} \circ \tilde{R} = \mathrm{Proj}_{E_q}$ is used to make the transfer

$$V_i^{(q)} := \tilde{R}^{<-1>}(V_i), \quad i \in [k].$$

One can easily check that the constructed subspaces $V_i^{(q)}$ for $i \in [k]$ are i.i.d. uniform l_q-dimensional subspaces of E_q, which is consistent with the setup in the original Game 9.4: using this construction instead of resampling probing subspaces is stochastically equivalent. Note that the run of Procedure 9.3 stops either when k depth-q exploratory queries were found, or a period of larger depth was completed. As a result, during lines 14-18 of Algorithm 9.5, one only needs to construct at most k depth-q probing subspaces. In summary,

Input: Period index J, Depth $q \in [p]$; dimensions d, $\tilde{d}$, $l_1, \ldots, l_P$; number of exploratory queries k; maximum number of queries T_{max}; algorithm alg for Game 9.4

Part 1: Initializing run of Procedure 9.3 conditionally on $\mathcal{E}_q(J)$

EventNotSatisfied $\leftarrow$**true**
while EventNotSatisfied **do**
  With fresh randomness, sample independently $E_1, \ldots, E_P$, uniform $\tilde{d}$-dim. subspaces in $\mathbb{R}^d$ and for $i \in [P-p]$ sample k uniform l_{p+i}-dim. subspaces of E_{p+i}: $V_j^{(p+i)}$ for $j \in [k]$
  Given all previous information, set the memory of alg to M-bit message Message and set $n_{p+i} \in [k]$ and vectors $\boldsymbol{y}_j^{(p+i)}$ for $j \in [n_{p+i}]$, for all $i \in [P-p]$; as in Game 9.4
  Set $n_{p'} \leftarrow 0$ for $p' \in [p]$ and reset subspaces $V_j^{(p+i)}$ for $i \in [P-p]$ and $j > n_{p+i}$
  for $t \in [T_{max}]$ **do**
    Run alg with current memory to get $\boldsymbol{x}_t$. Update exploratory queries $\boldsymbol{y}_i^{(p')}$ and probing subspaces $V_i^{(p')}$ for $p' \in [P], i \in [n_{p'}]$ with fresh randomness as in Procedure 9.3, and **return** $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg. **if** *n_p was reset because a deeper period was completed* **then break**;
    if *n_q was reset for the J-th time* **then**
      Rewind the state of the Procedure 9.3 variables to the exact moment when n_q was reset for the J-th time, in particular, before resampling $V_1^{(q)}$
      EventNotSatisfied $\leftarrow$**false** and $T \leftarrow t$; **break**
  end
end

Part 2: Continue the J-th depth-q period using oracle probing subspaces

Let $R : \mathbb{R}^d \to \mathbb{R}^d \in \mathcal{O}_d(\mathbb{R})$ be an isometry such that $R(E_q) = \mathbb{R}^{\tilde{d}} \otimes \{0\}^{d-\tilde{d}}$. Denote $\tilde{R} = \pi_{\tilde{d}} \circ R$ which only keeps the first $\tilde{d}$ coordinates of $R(\cdot)$, and let $\tilde{R}^{<-1>} : \mathbb{R}^{\tilde{d}} \to \mathbb{R}^d$ be the linear map for which $\tilde{R}^{<-1>} \circ \tilde{R} = \mathrm{Proj}_{E_q}$
for $t \in \{T, \ldots, T_{max}\}$ **do**
  if $t \neq T$ **then** Run alg with current memory to get $\boldsymbol{x}_t$;
  Update (For $t = T$, continue updating) exploratory queries $\boldsymbol{y}_i^{(p')}$ and probing subspaces $V_i^{(p')}$ for $p' \in [P], i \in [n_{p'}]$ as in Procedure 9.3, and **return** $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg. Whenever needed to sample a new depth-q probing subspace $V_i^{(q)}$ for $i \in [k]$, submit vector $\boldsymbol{y} = \tilde{R}(\boldsymbol{x}_t)$ to the oracle. Define $V_i^{(q)} := \tilde{R}^{<-1>}(V_i)$ where V_i is the oracle response. **if** *n_p was reset because a deeper period was completed* **then break**;
  if *n_q was reset during this iteration (depth-q period completed)* **then break**;
end
if *not all k queries to oracle were performed* **then** Query $\boldsymbol{y} = \boldsymbol{0}$ for remaining queries;

Algorithm 9.5: Strategy of the Player for the Probing Game 9.1

the algorithm never runs out of queries for Game 9.1, and queries during the run are stochas-

tically equivalent to those from playing the initial strategy on Game 9.4.

As a result, we can apply Theorem 9.5 to the strategy from Algorithm 9.5. It shows that on an event $\mathcal{E}$ of probability $1 - 4de^{-l_q/16}$, the strategy loses at the Probing Game 9.1 for parameter $\rho_q := \frac{1}{12}\sqrt{\frac{l_q}{\tilde{d}k^\alpha}}$. We now show that on the corresponding event $\tilde{\mathcal{E}}$ (replacing the construction of the probing subspaces using oracle subspaces line 16 of Algorithm 9.5 by a fresh uniform sample in l_q-dimensional subspaces of E_q), the depth-q period of Game 9.4 is proper. Indeed, note that in Part 2 of Algorithm 9.5, the times t when one queries the oracle are exactly depth-q exploration times. Let $\hat{n}_q$ be the total number of depth-q exploratory times during the run. The queries are exactly $\boldsymbol{y}_i := \tilde{R}(\boldsymbol{y}_i^{(q)})$ for $i \in [\hat{n}_q]$. On $\tilde{\mathcal{E}}$, for any $i \in [\hat{n}_q]$, there are no vectors $\boldsymbol{z} \in Span(\boldsymbol{y}_j, j \in [i])$ such that $\|\boldsymbol{z}\| = 1$ and $\|\mathrm{Proj}_{Span(V_j, j\in[i])}(\boldsymbol{z})\| \le \rho_q$. Applying the mapping $\tilde{R}^{<-1>}$ shows that for any vectors $\boldsymbol{z} \in Span(\mathrm{Proj}_{E_q}(\boldsymbol{y}_j^{(q)}), j \in [i])$, with $\|\boldsymbol{z}\| = 1$, we have

$$\|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])}(\boldsymbol{z})\| \ge \rho_q.$$

In other words,

$$\rho_q\|\boldsymbol{z}\| \le \|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])}(\boldsymbol{z})\|, \quad \boldsymbol{z} \in Span(\mathrm{Proj}_{E_q}(\boldsymbol{y}_j^{(q)}), j \in [i]). \tag{9.23}$$

Now consider any depth-p' query $\boldsymbol{x}_t$ with $p' > q$, during the interval of time between the i-th depth-q exploratory query and the following one. By definition (see Eq (9.6)), it passed all probes $V_1^{(q)}, \ldots, V_i^{(q)}$. That is,

$$\|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])}(\boldsymbol{x}_t)\| \le \delta_q. \tag{9.24}$$

We also have $\boldsymbol{e}^\top \boldsymbol{x}_t \le -\frac{1}{2}$ which implies $\|\boldsymbol{x}_t\| \ge \frac{1}{2}$. Next, either $\boldsymbol{x}_t$ was a depth-q exploratory query, in which case, $\boldsymbol{x}_t = \boldsymbol{y}_i^{(q)}$; or $\boldsymbol{x}_t$ does not satisfy the robustly-independent condition Eq (9.5) (otherwise $\boldsymbol{x}_t$ would be a depth-q exploratory query), that is

$$\|\mathrm{Proj}_{Span(\boldsymbol{y}_j^{(q)}, j\in[i])^\perp}(\boldsymbol{x}_t)\| < \delta_q. \tag{9.25}$$

In both cases, Eq (9.25) is satisfied. For convenience, denote $\boldsymbol{u} := \mathrm{Proj}_{Span(\boldsymbol{y}_j^{(p)}, j\in[i])}(\boldsymbol{x}_t)$. Applying Eq (9.23) with $\boldsymbol{z} = \mathrm{Proj}_{E_q}(\boldsymbol{u})$ yields

$$\begin{aligned}
\rho_q\|\mathrm{Proj}_{E_q}(\boldsymbol{u})\| &\le \|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])} \circ \mathrm{Proj}_{E_q}(\boldsymbol{u})\| \\
&= \|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])}(\boldsymbol{u})\| \\
&\le \|\boldsymbol{u} - \boldsymbol{x}_t\| + \|\mathrm{Proj}_{Span(V_j^{(q)}, j\in[i])}(\boldsymbol{x}_t)\| \\
&\le \delta_q + \delta_q = 2\delta_q
\end{aligned}$$

In the last inequality, we used Eq (9.24) and (9.25). As a result, on $\tilde{\mathcal{E}}$, all depth-p' queries $\boldsymbol{x}_t$ with $p' > q$ satisfy

$$\|\mathrm{Proj}_{E_q}(\boldsymbol{x}_t)\| \le \|\mathrm{Proj}_{E_q}(\boldsymbol{u})\| + \|\boldsymbol{x}_t - \boldsymbol{u}\| \le \frac{2\delta_q}{\rho_q} + \delta_p \le \frac{3\delta_q}{\rho_q} \le \eta_q.$$

In the last inequality, we used the definition of δ_q from Eq (9.9). This shows that under $\tilde{\mathcal{E}}$, the J-th depth-q period was proper. We recall that the J-th depth-q run was generated conditionally on $\mathcal{E}_q(J)$ (see Part 1 of Algorithm 9.5). Hence, we proved the desired result

$$\mathbb{P}\left(J\text{-th depth-}q \text{ period is proper} \mid \mathcal{E}_q(J)\right) \geq \mathbb{P}(\tilde{\mathcal{E}}) \geq 1 - 4de^{-l_q/16}.$$

This ends the proof. ■

9.4.4 Query lower bounds for the Orthogonal Subspace Game

We next show that during a depth-p period for $p \geq 2$, provided that the player won and the depth-$(p-1)$ periods during that interval of time were *proper* (which will be taken care of via Lemma 9.2), then a memory-constrained algorithm needs to have received $\Omega(\tilde{d})$ vectors from the previous depth $p-1$. This uses techniques from previous works on memory lower bounds for such orthogonal vector games, starting from [Mar+22]. For our purposes, we need a specific variant of these games, which we call the Orthogonal Subspace Game 9.2. This simulates a generic run of Procedure 9.3 during a period at any given depth $p \geq 2$. For the sake of presentation, we recall its definition here.

Input: dimensions d, $\tilde{d}$; memory M; number of robustly-independent vectors k; number of queries m; parameters β, γ

1 *Oracle:* Sample a uniform $\tilde{d}$-dimensional subspace E in $\mathbb{R}^d$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m \overset{i.i.d.}{\sim} \mathcal{U}(S_d \cap E)$

2 *Player:* Observe E and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$, and store an M-bit message Message about these

3 *Oracle:* Send samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ to player

4 *Player:* Based on Message and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ only, return unit norm vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$

5 The player wins if for all $i \in [k]$

1. $\|\mathrm{Proj}_E(\boldsymbol{y}_i)\| \leq \beta$
2. $\|\mathrm{Proj}_{Span(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{i-1})^\perp}(\boldsymbol{y}_i)\| \geq \gamma$.

Game 9.2: Orthogonal Subspace Game

Our end goal is to prove a query lower bound $\Omega(\tilde{d})$ on Game 9.2 for the player to succeed with reasonable probability. To do so, we simplify the game further by deleting the queries $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ altogether. This yields Game 9.7.

Precisely, we show that a strategy to play the Orthogonal Subspace Game 9.2 yields a strategy for the Simplified Orthogonal Subspace Game 9.7 for the new dimension $d' = d - m$. The following lemma formalizes this reduction.

Lemma 9.3. *If there is an algorithm for the Orthogonal Subspace Game 9.2 with parameters $(d, \tilde{d}, M, k, m, \beta, \gamma)$, then there is a strategy for the Simplified Subspace Game 9.7 with parameters $(d, \tilde{d} - m, M, k, \beta, \gamma)$ that wins with at least the same probability.*

Proof Fix a strategy for the Orthogonal Subspace Game 9.2. We define in Algorithm 9.8 a strategy for Game 9.7 for the desired parameters $(d, \tilde{d} - m)$.

Input: dimensions d, $\tilde{d}$; memory M; number of vectors k; parameters β, γ

1 *Oracle:* Sample a uniform $\tilde{d}$-dimension linear subspace E in $\mathbb{R}^d$

2 *Player:* Observe E and store an M-bit message Message about E

3 *Player:* Based on Message only, return unit norm vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$

4 The player wins if for all $i \in [k]$

1. $\|\mathrm{Proj}_E(\boldsymbol{y}_i)\| \leq \beta$
2. $\|\mathrm{Proj}_{Span(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{i-1})^\perp}(\boldsymbol{y}_i)\| \geq \gamma$.

Game 9.7: Simplified Orthogonal Subspace Game

The intuition is the following: E is sampled as a uniform $\tilde{d}$-dimensional subspace of $\mathbb{R}^d$. On the other hand, if V (resp. F') is a uniform m-dimensional (resp. $(\tilde{d}-m)$-dimensional) subspace of $\mathbb{R}^d$ then the subspace $V \oplus F'$ is also distributed as a uniform $\tilde{d}$-dimensional subspace of $\mathbb{R}^d$. Now note that the subspace F from the oracle of Game 9.7 is distributed exactly as a $(\tilde{d}-m)$-dimensional subspace of $\mathbb{R}^d$. Therefore, we can simulate the subspace E from Game 9.2 via $E = V \oplus F$. It remains to simulate $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$. To do so, note that conditionally on $E := F \oplus V$, the subspace V is a uniformly random m-dimensional subspace of E. Similarly, for i.i.d. sampled vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m \overset{i.i.d.}{\sim} \mathcal{U}(S_{d-1} \cap E)$, the space $Span(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$ is also a uniform m-dimensional subspace of E (on an event $\mathcal{E}$ of probability one). Last, for any m-dimensional subspace V , denote by $\mathcal{D}(V)$ the conditional distribution of $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$ conditionally on $Span(\boldsymbol{w}_i, i \in [m]) = V$. (Note that $\mathcal{D}(V)$ does not correspond to $\mathcal{U}(S_d \cap V)^{\otimes m}$ because the vectors $\boldsymbol{v}_i$ will tend to be more "orthogonal" since they were initially sampled in a $\tilde{d}$-dimensional subspace E while V has only dimension m). As a summary of the previous discussion, by sampling vectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) \sim \mathcal{D}(V)$, conditionally on $E = V \oplus F$, these are distributed exactly as i.i.d. uniform $\mathcal{U}(S_{d-1} \cap E)$ samples. We can then use the following procedure to sample E and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$:

1. Let F be the $(\tilde{d}-m)$-dimensional subspace provided by the oracle of Game 9.7.
2. Sample a uniform m-dimensional subspace V of $\mathbb{R}^d$. Sample $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) \sim \mathcal{D}(V)$ and define $E = V \oplus F$.

The procedure is stochastically equivalent to the setup line 1 of Game 9.2. The complete strategy for the simplified Game 9.7 is given in Algorithm 9.8.

Now suppose that the player won at Game 9.2. Then, the outputs $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ are normalized and satisfy the desired robust-independence property. Further, for all $i \in [k]$, we have

$$\|\mathrm{Proj}_F(\boldsymbol{y}_i)\| \leq \|\mathrm{Proj}_E(\boldsymbol{y}_i)\| \leq \beta.$$

Hence, Algorithm 9.8 wins at Game 9.7 on the same event. ■

In Game 9.7, provided that the message does not contain enough information to store the outputs $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ directly, we will show that the player cannot win with significant probability. This should not be too surprising at this point, because the message is all the player has access to output vectors that are roughly orthogonal to the complete space E.

Input: dimensions d, $\tilde{d}$; memory M; number of robustly-independent vectors k; number of queries m; strategy for Game 9.2

Part 1: Construct Message

Receive F a $(\tilde{d}-m)$-dimensional subspace provided by the oracle

Sample a uniform m-dimensional subspace V of $\mathbb{R}^d$. Sample $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_m) \sim \mathcal{D}(V)$ and define $E = V \oplus F$. Store message Message given E and $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ as in Game 9.2

Part 2: Output solution vectors

Observe Message and resample V, $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ using the same randomness as in Part 1

return $\boldsymbol{y}_1, \dots, \boldsymbol{y}_k$, the same outputs given by the strategy for Game 9.2

Algorithm 9.8: Strategy for the Simplified Game 9.7 given a strategy for Game 9.2

Lemma 9.4. *Let $d \geq 8$, $k \leq \frac{\tilde{d}}{2}$ and $0 < \beta, \gamma \leq 1$ such that $\gamma/\beta \geq 3e\sqrt{kd/\tilde{d}}$. Then, the success probability of a player for Game 9.7 for memory M satisfies*

$$\mathbb{P}(\textit{player wins}) \leq 25 \cdot \frac{M+2}{\tilde{d}k} \ln \frac{\sqrt{d}}{\gamma}.$$

For this, we use a lemma that constructs an orthonormal sequence of vectors from robustly-independent vectors. Versions of this property were already observed in [Mar+22, Lemma 34], or Lemma 7.6 from Chapter 7. To obtain query lower bounds that reach the query complexity of gradient descent $1/\epsilon^2$, we need the tightest form of that result. The proof is included in appendix.

Lemma 9.5. *Let $\delta \in (0, 1]$ and $\boldsymbol{y}_1, \dots, \boldsymbol{y}_r \in \mathbb{R}^d$ some $r \leq d$ unit norm vectors. Suppose that for any $i \leq k$,*

$$\|P_{Span(\boldsymbol{y}_j, j<i)^\perp}(\boldsymbol{y}_i)\| \geq \delta.$$

Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_r]$ and $s \geq 2$. There exists $\lceil r/s \rceil$ orthonormal vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_{\lceil r/s \rceil}]$ such that for any $\boldsymbol{a} \in \mathbb{R}^d$,

$$\|\boldsymbol{Z}^\top \boldsymbol{a}\|_\infty \leq \left(\frac{\sqrt{r}}{\delta}\right)^{s/(s-1)} \|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty.$$

Further, these can be constructed as the singular vectors of the singular value decomposition of $\boldsymbol{Y}$ associated with the $\lceil r/s \rceil$ largest singular values.

We are now ready to prove Lemma 9.4.

Proof of Lemma 9.4 Fix a strategy for Game 9.7 and $s = 1 + \ln \frac{\sqrt{d}}{\gamma}$. For simplicity, without loss of generality assume that the message Message is deterministic in E (by the law of total probability, there is a choice of internal randomness such that running the strategy with that randomness yields at least the same probability of success). Now denote by $\mathcal{E}$ the event when the player wins and let $\boldsymbol{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_k]$ be the concatenation of the vectors output by the player. By Lemma 9.5, we can construct an orthonormal sequence $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_r]$

with $r = \lceil k/s \rceil$ such that on the event $\mathcal{E}$, for all $i \in [r]$ and $\boldsymbol{a} \in E$ with unit norm,

$$\|\boldsymbol{z}_i^\top \boldsymbol{a}\| \leq \left(\frac{\sqrt{k}}{\gamma}\right)^{s/(s-1)} \|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty \leq \left(\frac{\sqrt{k}}{\gamma}\right)^{s/(s-1)} \max_{j\in[k]} \|\mathrm{Proj}_E(\boldsymbol{y}_j)\|$$
$$\leq \beta \left(\frac{\sqrt{k}}{\gamma}\right)^{1+\frac{1}{s-1}} \leq \frac{e\beta\sqrt{k}}{\gamma}.$$

In other words, we have for all $i \in [r]$,

$$\|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \leq \frac{e\beta\sqrt{k}}{\gamma} \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}}. \tag{9.26}$$

We now give both an upper and lower bound on $I(E, \boldsymbol{Y})$ – this will lead to the result. First, because $\boldsymbol{Y}$ is constructed from $\mathsf{Message}$, the data processing inequality gives

$$I(E; \boldsymbol{Y}) \leq I(E; \mathsf{Message}) \leq H(\mathsf{Message}) \leq M \ln 2. \tag{9.27}$$

To avoid continuous/discrete issues with the mutual information, we provide the formal justification of the computation above. For any $\mathcal{M} \in \{0,1\}^M$, let $\mathcal{S}(\mathcal{M}) = \{E : \mathsf{Message}(E) = \mathcal{M}\}$ the set of subspaces that lead to message $\mathcal{M}$. This is well defined because we supposed $\mathsf{Message}$ to be deterministic in E. Because $\boldsymbol{Y}$ depends only on $\mathsf{Message}$, we also define $p_{\boldsymbol{Y},\mathcal{M}}$ the probability mass function for $\boldsymbol{Y}$ given message $\mathcal{M}$, and $p_M(\mathcal{M})$ the probability to have $\mathsf{Message} = \mathcal{M}$. Then,

$$\begin{aligned}
I(E; \boldsymbol{Y}) &:= \int_e \int_{\boldsymbol{y}} p_{E,\boldsymbol{Y}}(e, \boldsymbol{y}) \ln \frac{p_{E,\boldsymbol{Y}}(e, \boldsymbol{y})}{p_E(e) p_{\boldsymbol{Y}}(\boldsymbol{y})} de d\boldsymbol{y} \\
&= \sum_{\mathcal{M}\in\{0,1\}^M} \int_{e\in\mathcal{S}(\mathcal{M})} \int_{\boldsymbol{y}} p_E(e) p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y}) \ln \frac{p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} de d\boldsymbol{y} \\
&= \sum_{\mathcal{M}\in\{0,1\}^M} p_M(\mathcal{M}) \int_{\boldsymbol{y}} p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y}) \ln \frac{p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} d\boldsymbol{y} \\
&= \int_{\boldsymbol{y}} d\boldsymbol{y} \sum_{\mathcal{M}\in\{0,1\}^M} p_M(\mathcal{M}) p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y}) \ln \frac{p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y})}{\sum_{\mathcal{M}'\in\{0,1\}^M} p_M(\mathcal{M}') p_{\boldsymbol{Y},\mathcal{M}'}(\boldsymbol{y})} \\
&\leq \int_{\boldsymbol{y}} d\boldsymbol{y} \sum_{\mathcal{M}\in\{0,1\}^M} p_M(\mathcal{M}) p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y}) \ln \frac{1}{p_M(\mathcal{M})} = H(\mathcal{M}) \leq M \ln 2.
\end{aligned}$$

In the last inequality, we used the standard inequality $H(X) \geq 0$ where X is the discrete random variable with $p_X(\mathcal{M}) \propto p_M(\mathcal{M}) p_{\boldsymbol{Y},\mathcal{M}}(\boldsymbol{y})$.

We now turn to the lower bound. Recall that from Lemma 9.5, the vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r$ are constructed explicitely from $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k]$ as the vectors from the r largest singular values for the singular value decomposition of $\boldsymbol{Y}$. As a result, by the data processing inequality, we have

$$I(E; \boldsymbol{Y}) = I(E; \boldsymbol{Y}, \boldsymbol{Z}) \geq I(E; \boldsymbol{Z}).$$

By the chain rule,

$$\begin{aligned} I(E;\boldsymbol{Z}) &= I(E;\boldsymbol{Z},\mathbb{1}[\mathcal{E}]) + I(E;\mathbb{1}[\mathcal{E}]) - I(E;\mathbb{1}[\mathcal{E}] \mid \boldsymbol{Z}) \\ &\geq I(E;\boldsymbol{Z} \mid \mathbb{1}[\mathcal{E}]) - H(\mathbb{1}[\mathcal{E}]) \\ &\geq \mathbb{P}(\mathcal{E})\mathbb{E}_{\mathcal{E}}\left[I(E;\boldsymbol{Z} \mid \mathcal{E})\right] - \ln 2. \end{aligned}$$

Now fix a possible realization for $\boldsymbol{Z}$. We recall that provided that the player won ($\mathcal{E}$ is satisfied), the matrix $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, z_r]$ satisfies Eq (9.26). Let $\mathcal{C}$ denote the set of all subspaces E compatible with these:

$$\mathcal{C} := \mathcal{C}(\boldsymbol{Z}) = \left\{ \tilde{d}\text{-dimensional subspace } F \text{ of } \mathbb{R}^d : \|\mathrm{Proj}_F(\boldsymbol{z}_i)\| \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}}, i \in [r] \right\}.$$

This set is measurable as the intersection of measurable sets. By the data processing inequality,

$$I(E;\boldsymbol{Z} \mid \mathcal{E}) \geq I(E;\mathcal{C} \mid \mathcal{E}) = \mathbb{E}_{\mathcal{C}|\mathcal{E}}\left[D(p_{E|\mathcal{C},\mathcal{E}} \parallel p_{E|\mathcal{E}})\right]$$

Note that $p_{E|\mathcal{E}} = p_{E,\mathcal{E}}/\mathbb{P}(\mathcal{E}) \leq p_E/\mathbb{P}(\mathcal{E})$. Since $\mathcal{E}$ is satisfied, the support of $p_{E|\mathcal{C},\mathcal{E}}$ is included in $\mathcal{C}$. Now let F be a uniformly sampled subspace of $\mathcal{C}$. We obtain

$$D(p_{E|\mathcal{C},\mathcal{E}} \parallel p_{E|\mathcal{E}}) = \mathbb{E}_{E|\mathcal{C},\mathcal{E}}\left[\ln \frac{p_{E|\mathcal{C},\mathcal{E}}}{p_{E|\mathcal{E}}}\right] \geq \mathbb{E}_{E|\mathcal{C},\mathcal{E}}\left[\ln \frac{\mathbb{P}(\mathcal{E})p_{F|\mathcal{C}}}{p_E}\right] = \ln \frac{1}{\mathbb{P}(E \in \mathcal{C})} + \ln \mathbb{P}(\mathcal{E}). \quad (9.28)$$

The last step of the proof is to upper bound this probability $\mathbb{P}(E \in \mathcal{C})$ where here $\mathcal{C}$ and $\boldsymbol{Z}$ were fixed. Without loss of generality (since the distribution of E is rotation-invariant), we can assume that $\boldsymbol{z}_i = \boldsymbol{e}_i$ for $i \in [r]$, the first r vectors of the natural basis of $\mathbb{R}^d$. Equivalently, we can consider the setup where $E = \mathbb{R}^{\tilde{d}} \otimes \{0\}^{d-\tilde{d}}$ and we sample a random orthonormal sequence $\boldsymbol{Z}$ of r vectors uniformly in $\mathbb{R}^d$, which does not affect the quantity

$$\mathbb{P}(E \in \mathcal{C}) = \mathbb{P}\left(\|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}},\ i \in [r]\right).$$

We take this perspective from now. For $i \in [r]$, we introduce $G_i = \mathit{Span}(\boldsymbol{z}_j, \mathrm{Proj}_E(\boldsymbol{z}_j), j \in [i])$. We then define

$$F_i = E \cap G_{i-1}^{\perp} \quad \text{and} \quad \boldsymbol{a}_i = \mathrm{Proj}_{G_{i-1}^{\perp}}(\boldsymbol{z}_i).$$

In particular, we have $E = \mathit{Span}(\mathrm{Proj}_E(\boldsymbol{z}_j), j < i) \oplus F_i$. Recall that because $\boldsymbol{z}_1, \dots, \boldsymbol{z}_r$ was sampled as a uniformly rotated orthonormal sequence, conditionally on $\boldsymbol{z}_j$ for $j < i$ (and also on $\mathrm{Proj}_E(\boldsymbol{z}_j)$ for $j < i$, which do not bring further information on $\boldsymbol{z}_i$), the variable $\boldsymbol{z}_i$ is exactly distributed as a random uniform unit vector in $\mathit{Span}(\boldsymbol{z}_j, j < i)^{\perp}$. Within this space is included $F_i \subset G_{i-1}^{\perp}$, hence we can apply Lemma 9.14 to obtain

$$\mathbb{P}\left(\|\mathrm{Proj}_{F_i}(\boldsymbol{z}_i)\| \leq \sqrt{\frac{\dim(F_i)}{d-i+1}\left(1 - \frac{1}{\sqrt{2}}\right)} \mid \boldsymbol{z}_j, \mathrm{Proj}_E(\boldsymbol{z}_j), j < i\right) \leq e^{-\dim(F_i)/8}.$$

Because $r \leq k \leq \frac{\tilde{d}}{4}$, we have $\dim(F_i) \geq \dim(E) - \dim(G_{i-1}) \geq \tilde{d} - 2(r-1) \geq \frac{\tilde{d}}{2}$. Hence, the previous equation implies

$$\mathbb{P}\left(\|\boldsymbol{a}_i\| \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}} \mid \boldsymbol{z}_j, \mathrm{Proj}_E(\boldsymbol{z}_j), j < i \right) \leq e^{-\tilde{d}/16}.$$

Combining this equation together with the fact that

$$\|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \geq \|\mathrm{Proj}_{F_i}(\boldsymbol{z}_i)\| = \|\mathrm{Proj}_{F_i}(\boldsymbol{a}_i)\|,$$

we obtained

$$\mathbb{P}\left(\|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}} \mid \boldsymbol{z}_j, \mathrm{Proj}_E(\boldsymbol{z}_j), j < i \right) \leq e^{-\tilde{d}/16}.$$

Because this holds for all $i \in [r]$, this implies that

$$\mathbb{P}(E \in \mathcal{C}(\boldsymbol{Z})) \leq e^{-\tilde{d}r/16}.$$

We can then plug this bound into Eq (9.28). This finally yields

$$M \ln 2 \geq I(E; \boldsymbol{Y}) \geq \mathbb{P}(\mathcal{E})\frac{\tilde{d}r}{16} - \mathbb{P}(\mathcal{E}) \ln \frac{1}{\mathbb{P}(\mathcal{E})} - \ln 2 \geq \mathbb{P}(\mathcal{E})\frac{\tilde{d}k}{16s} - \ln 2 - \frac{1}{e}.$$

In the left inequality, we recalled the information upper bound from Eq (9.27). Because we assumed $d \geq 8$, we have $s \leq 2 \ln \frac{\sqrt{d}}{\gamma}$. Rearranging and simplifying ends the proof. ■

As a result, combining the reduction from the Orthogonal Subspace Game 9.2 to the simplified Game 9.7 from Lemma 9.3, together with the query lower bound in Lemma 9.4, we obtain the desired query lower bound for Game 9.2.

Proof of Theorem 9.4 The proof essentially consists of putting together Lemmas 9.3 and 9.4. Suppose that the player uses at most $m < \frac{\tilde{d}}{2}$ queries. By Lemma 9.3, we can use this strategy to solve Game 9.7, where the dimension of the subspace E is $\dim(E) = \tilde{d} - m > \frac{\tilde{d}}{2}$. Using this bound, we can check that the parameters satisfy the conditions from Lemma 9.4, that is

$$k \leq \frac{\dim(E)}{2} \quad \text{and} \quad \frac{\gamma}{\beta} \geq 3e\sqrt{\frac{kd}{\dim(E)}}.$$

As a result, we have

$$\mathbb{P}(\text{player wins}) \leq 25\frac{M+2}{\dim(E)k} \ln \frac{\sqrt{d}}{\gamma} < 50\frac{M+2}{\tilde{d}k} \ln \frac{\sqrt{d}}{\gamma} \leq \frac{1}{C}.$$

This gives a contradiction and ends the proof. ■

9.4.5 Recursive query lower bounds for the feasibility game

It remains to relate the Orthogonal Subspace Game 9.2 to the feasibility Game 9.4 to obtain query lower bounds for the latter using Theorem 9.4. We briefly give some intuition as to how this reduction works. Intuitively, during a run of a period of depth p from Game 9.4 the algorithm needs to find k exploratory queries that are by definition robustly-independent (Eq (9.5)). Using Lemma 9.2 we also show that most of the periods at depth $p-1$ are proper, hence the exploratory queries also need to be roughly orthogonal to E_{p-1}. We can therefore emulate a run of Game 9.2 by taking $E_{p-1} = E$ to be the hidden random subspace. This gives the following result.

Lemma 9.6. *Let $p \in \{2, \dots, P\}$. Suppose that $\frac{\tilde{d}}{4l} \geq k \geq 50 \cdot (4P)^{\frac{M+\frac{d}{8}\log_2(T_{max})+3}{\tilde{d}}} \ln \frac{\sqrt{d}}{\delta_p}$. If there exists a strategy for Game 9.4 for depth p that wins with probability at least q, then during a run of the strategy,*

$$\mathbb{P}\left(\textit{at least } \frac{\tilde{d}}{4lk} \textit{ periods of depth } p-1 \textit{ are completed}\right) \geq q - \frac{3}{8P}.$$

Proof Fix $p \in \{2, \dots, P\}$ and a strategy for the Depth-p Game 9.4. We construct in Algorithm 9.9 a strategy for Game 9.2 for $m = \left\lceil \tilde{d}/2 \right\rceil - 1$. It simulates a run of Game 9.4 by sampling subspaces $E_{p'}$ for $p' \neq p-1$ and using for E_{p-1} the subspace E sampled by the oracle. The message Message, constructed in Part 1 on Algorithm 9.9, contains the initialization for the memory of the underlying algorithm alg, as well as indications of when are the exploratory queries for periods of depth $p' > p$. In Part 2 of Algorithm 9.9, the run of Game 9.4 is simulated once again, but without having direct access to E. Fortunately, to compute the feasibility separation oracle from Game 9.4 (or Procedure 9.3), one only needs to:

1. Construct uniformly sampled subspaces $V_i^{(p-1)}$ of $E = E_{p-1}$. This can be done directly thanks to the vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ provided by the oracle in line 3 of Game 9.2. Indeed, for any vectors $\boldsymbol{z}_1, \dots, \boldsymbol{z}_l \overset{i.i.d.}{\sim} \mathcal{U}(S_d \cap E)$, the distribution of $Span(\boldsymbol{z}_1, \dots, \boldsymbol{z}_l)$ is the same as for a uniformly sampled l-dimensional subspace of $E_{p-1} = E$. We therefore use l new vectors within the list $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ whenever a new probing subspace of E_{p-1} is needed. (Recall that $l_{p-1} = l$ since $p-1 < P$.)

2. Know when queries are exploratory queries. This is important to update the number of exploratory queries $n_{p'}$ for $p' \in [P]$ which dictates the number of probing subspaces needed. For $p' \in [p]$, this can be done directly since all depth-p' periods start with no exploratory queries ($n_{p'} \leftarrow 0$ in line 3 of Game 9.4). Hence all previous depth-p' exploratory queries are queried during the run Part 2, and we can test for robust-independence (Eq (9.5)) directly. This is more problematic for depths $p' > p$ because these depend on the vectors $\boldsymbol{y}_j^{(p')}$ for $j \in [n_{p'}]$ defined in line 2 of Game 9.4 with knowledge of $E_{p-1} = E$. These contain too many bits to be included in Message. Fortunately, we only need to store the times of these depth-p' exploratory queries, which sidesteps checking for robust-independence (Eq (9.5)). To know when these

times occur, we need to simulate the complete Game 9.4 in Part 1, lines 3-11 of Algorithm 9.9, also using the oracle samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ that would be used in Part 2 to construct depth-$(p-1)$ probing subspaces.

We then define the message as $\mathsf{Message} = (\mathsf{Memory}; t_j^{(p+i)}, i \in [P-p], j \in [k])$ where $t_j^{(p+i)}$ is the time of the j-th exploratory query of depth $p+i$, with the convention $t_j^{(p+i)} = -1$ if there were no j depth-$(p+i)$ exploratory queries (line 12 of Algorithm 9.9). With this convention, we can also know exactly what was n_{p+i} for $i \in [P-p]$ from the message, and it can be stored with a number of bits of

$$M + \lceil (P-p)k \log_2(T_{max}+2) \rceil \leq M + \frac{d}{8} \log_2(T_{max}+2) + 1.$$

Here we used $Pk \leq P\tilde{d} \leq \frac{d}{2}$. In Part 2, after simulating the run from Game 9.4, the strategy returns the (normalized) depth-p exploratory queries to the oracle (line 19).

We now show that this strategy wins with significant probability. Note that a run of Game 9.4 ends whenever the depth-p period is finished. In particular, this ensures that there is no overwriting of the depth-p exploratory queries, hence, there is no ambiguity when referring to some exploratory query $\boldsymbol{y}_i^{(p)}$. We next let $\mathcal{E}$ be the event when the strategy for Game 9.4 wins. Because E is also sampled uniformly as a $\tilde{d}$-dimensional subspace by the oracle, under the corresponding event $\tilde{\mathcal{E}}$ (only changing the dependency in E_{p-1} by the dependency in E), the strategy succeeds, that is, the algorithm makes k depth-p exploratory queries.

We apply Lemma 9.2 checking that the assumptions are satisfied: $l \geq C_\alpha \ln n$ from Eq (9.7) and $4lk \leq \tilde{d}$, where $\tilde{d}$ is the dimension of the problem for Game 9.1 here. Hence, by the union bound, on an event $\mathcal{F}$ of probability at least $1 - 4d(k)de^{-l/16}$, the first $d(k) := \left\lfloor \frac{\tilde{d}}{k} \right\rfloor$ periods of depth $p-1$ are proper. Indeed,

$$\begin{aligned}
\mathbb{P}(\mathcal{F}^c) &= \mathbb{P}(\exists j \in [d(k)] : j\text{-th depth-}(p-1)\text{ period was started and is not proper}) \\
&\leq \sum_{j \in [d(k)]} \mathbb{P}(\mathcal{E}_{p-1}(j) \cap \{j\text{-th depth-}(p-1)\text{ period is not proper}\} \\
&\leq \sum_{j \in [d(k)]} 4de^{-l/16} \mathbb{P}(\mathcal{E}_{p-1}(j)) \leq 4d^2 e^{-l/16}.
\end{aligned}$$

Last, let $\mathcal{G}$ be the event that there are at most $\frac{m}{lk}$ periods of depth $p-1$. On each period of depth $p-1$, Algorithm 9.9 only uses at most lk samples $\boldsymbol{v}_i$ from the oracle: k for each probing subspace $V_i^{(p-1)}$. We note that because the algorithm stops as soon as a depth-p' period for $p' \geq p$ ends, the probing subspaces $V_i^{(p-1)}$ are never reset because deeper periods ended (see line 12 of Procedure 9.3). Thus, on $\mathcal{G}$, Algorithm 9.9 does not run out of oracle samples.

On $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, all k depth-p exploratory queries were made during proper periods of depth $p-1$. Here we used the fact that on $\mathcal{G}$, there are at most $\frac{m}{lk} \leq d(k)$ periods of depth $p-1$. By definition of proper periods (Definition 9.3), we have

$$\|\mathrm{Proj}_{E_{p-1}}(\boldsymbol{y}_i^{(p)})\| \leq \eta_{p-1}, \quad i \in [k].$$

Input: depth p, dimensions d, $\tilde{d}$, number of vectors k, M-bit algorithm alg for Game 9.4 at depth p; T_{max}

Part 1: Constructing the message

1 Sample independently $E_{p'}$ for $p' \in [P] \setminus \{p-1\}$, uniform $\tilde{d}$-dimensional linear subspaces in $\mathbb{R}^d$ and for $i \in [P-p]$ sample k uniform l-dimensional subspaces of E_{p+i}: $V_j^{(p+i)}$ for $j \in [k]$

2 Observe E and set $E_{p-1} = E$. Given all previous information, set the memory of alg to M-bit message Memory and set $n_{p+i} \in [k]$ and vectors $\boldsymbol{y}_j^{(p+i)}$ for $j \in [n_{p+i}]$, for all $i \in [P-p]$; as in Game 9.4

3 Set $n_{p'} \leftarrow 0$ for $p' \in [p]$

4 Receive samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$, and set sample index $i \leftarrow 1$

5 **for** $t \in [T_{max}]$ **do**

6 Run alg with current memory to get $\boldsymbol{x}_t$. Update exploratory queries $\boldsymbol{y}_i^{(p')}$ and probing subspaces $V_i^{(p')}$ for $p' \in [P], i \in [n_{p'}]$ as in Procedure 9.3. If n_p was reset because a deeper period was completed, strategy fails: **end** procedure. Whenever needed to sample a depth-$(p-1)$ probing subspace $V_{n_{p-1}}^{(p-1)}$ of E_{p-1} (lines 9 or 12 of Procedure 9.3):

7 **if** $i + l - 1 > m$ **then** Strategy fails: **end** procedure;

8 **else** use oracle samples, set $V_{n_{p-1}}^{(p-1)} := Span(\boldsymbol{v}_i, \ldots, \boldsymbol{v}_{i+l-1})$ and $i \leftarrow i + l$;

9 **return** $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg

10 **if** $n_p = k$ **then break**;

11 **end**

12 For $i \in [P-p]$ denote $t_1^{(p+i)}, \ldots, t_{n_{p+i}}^{(p+i)}$ the times of depth-$(p+i)$ exploratory queries. If these were done before $t = 1$ set them to 0. Store message $\mathsf{Message} = (\mathsf{Memory}; t_j^{(p+i)}, i \in [P-p], j \in [k])$ (let $t_j^{(p+i)} = -1$ if $j > n_{p+i}$)

Part 2: Simulate run of Game 9.4

13 Receive samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ from Oracle

14 Resample $E_{p'}$ for $p' \in [P] \setminus \{p-1\}$ using same randomness as in Part 1. Initialize memory of alg to Memory. Set $n_{p'} \leftarrow 0$ for $p' \in [p]$, and sample index $i \leftarrow 1$

15 **for** $t \in [T_{max}]$ **do**

16 Run alg with current memory to get $\boldsymbol{x}_t$. Update exploratory queries and probing subspaces exactly as in line 6, with the same randomness as in Part 1 for sampling probing subspaces. To know whether $\boldsymbol{x}_t$ is a depth-$(p+i)$ exploratory query for $i \in [P-p]$ (line 6 of Procedure 9.3), check whether t is within the message times $t_j^{(p+i)}$ for $j \in [n_{p+i}]$

17 Return $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg. **if** $n_p = k$ **then break**;

18 **end**

19 **return** normalized depth-p exploratory queries $\frac{\boldsymbol{y}_1^{(p)}}{\|\boldsymbol{y}_1^{(p)}\|}, \ldots, \frac{\boldsymbol{y}_k^{(p)}}{\|\boldsymbol{y}_k^{(p)}\|}$

Algorithm 9.9: Strategy of the Player for the Orthogonal Subspace Game 9.2

Now recall that exploratory queries also satisfy $\boldsymbol{e}^\top \boldsymbol{y}_i^{(p)} \le -\frac{1}{2}$, so that $\frac{1}{2} \le \|\boldsymbol{y}_i^{(p)}\| \le 1$ for $i \in [k]$. As a result, the output normalized vectors $\boldsymbol{u}_i = \boldsymbol{y}_i^{(p)}/\|\boldsymbol{y}_i^{(p)}\|$ for $i \in [k]$ satisfy

$$\|\mathrm{Proj}_{E_{p-1}}(\boldsymbol{u}_i^{(p)})\| \le 2\|\mathrm{Proj}_{E_{p-1}}(\boldsymbol{y}_i^{(p)})\| \le 2\eta_{p-1}, \quad i \in [k].$$

Also, by construction exploratory queries are robustly-independent (Eq 9.5). Hence,

$$\|\mathrm{Proj}_{Span(\boldsymbol{u}_j^{(p)}, j<i)^\perp}(\boldsymbol{u}_i^{(p)})\| \ge \|\mathrm{Proj}_{Span(\boldsymbol{y}_j^{(p)}, j<i)^\perp}(\boldsymbol{y}_i^{(p)})\| \ge \delta_p, \quad j \in [k].$$

This shows that on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, the algorithm wins at Game 9.2 with memory at most $M + \frac{d}{8}\log_2(T_{max}+2)+1$, using $m < \tilde{d}/2$ queries and for the parameters $(\beta, \gamma) = (2\eta_{p-1}, \delta_p)$. We now check that the assumptions for applying Theorem 9.4 are satisfied. The identity $d \ge 8P$ is satisfied by assumption throughout the chapter. The only assumption that needs to be checked is that for γ/β. Using the notation $\mu_p = \mu$ for all $p \in [P-1]$, we now note that

$$\frac{\gamma}{\beta} = \frac{\delta_p}{2\eta_{p-1}} = \frac{\mu_p}{72}\sqrt{\frac{l_p}{\tilde{d}k^\alpha}} = \frac{600}{72}\sqrt{\frac{kd}{\tilde{d}}} \ge 12\sqrt{\frac{kd}{\tilde{d}}}.$$

Here we used the definitions Eq (9.8) and (9.9). The lower bound k from the hypothesis is exactly the bound needed to apply Theorem 9.4 with the probability $1/(4P)$. Precisely, we have

$$\mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \le \mathbb{P}(\text{Algorithm 9.9 wins at Game 9.2}) \le \frac{1}{4P}.$$

Combining the previous statements shows that

$$\begin{aligned}
\mathbb{P}\left(\text{more than } \frac{m}{lk} \text{ periods of depth } p-1\right) &= \mathbb{P}(\mathcal{G}^c) \\
&\ge \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}^c) = \mathbb{P}(\mathcal{E} \cap \mathcal{F}) - \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \\
&\ge \mathbb{P}(\mathcal{E}) - \mathbb{P}(\mathcal{F}^c) - \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \\
&\ge q - \frac{1}{4P} - 4d^2 e^{-l/16}.
\end{aligned}$$

Because we also have $l \ge 16\ln(32d^2P)$ from Eq (9.7), this shows that

$$\mathbb{P}\left(\text{more than } \frac{m}{lk} \text{ periods of depth } p-1\right) \ge q - \frac{3}{8P}.$$

In this event there are at least $\frac{\tilde{d}/2}{lk}$ periods of depth $p-1$, hence at least $\frac{\tilde{d}}{2lk} - 1 \ge \frac{\tilde{d}}{4lk}$ are complete. ■

We are now ready to state the main recursion lemma, which enables us to construct an algorithm for Game 9.4 at depth $p-1$ from an algorithm for depth p.

Lemma 9.7. *Let $p \in \{2, \dots, P\}$ and suppose that the assumptions on k from Lemma 9.6 are satisfied. Suppose that there is a strategy for Game 9.4 for depth p with $T_{max}^{(p)}$ queries, that uses M bits of memory and that wins with probability at least $q \in [0,1]$. Then, there is a strategy for Game 9.4 for depth $p-1$ with*

$$T_{max}^{(p-1)} := \frac{32Plk}{\tilde{d}} T_{max}^{(p)}$$

queries, that uses the same memory and wins with probability at least $q - \frac{1}{2P}$.

Proof Fix $p \in \{2, \dots, P\}$ and the strategy for depth p. By Lemma 9.6, on an event $\mathcal{E}$ of probability at least $q - \frac{3}{8P}$, there are at least $K_0 = \left\lceil \frac{\tilde{d}}{4lk} \right\rceil$ completed depth-$(p-1)$ periods.

The main remark is that given $E_1, \dots, E_p$, the separation oracle from Procedure 9.3 is defined exactly similarly for each new depth-$(p-1)$ period. This is the reason why we reset all information from depths $q \le p'$ whenever a period of depth p' ends (line 12 of Procedure 9.3). In fact, the distribution of outcomes for the run of a depth-$(p-1)$ period is completely characterized by the memory state at the beginning of that period, as well as the exploratory queries for depths $p' > p-1$. Under $\mathcal{E}$, in average, the depth-$(p-1)$ periods are completed using relatively few iterations, hence we aim to simulate a depth-$(p-1)$ run to solve Game 9.4 for depth $p-1$. The strategy for the player is as follows:

1. Draw an index $\mathsf{it} \sim \mathcal{U}([K_0])$.
2. Run the strategy for depth p using the separating oracle from Procedure 9.3 until the beginning of the it-th period of depth $p-1$. If the procedure never finishes $\mathsf{it}-1$ depth-$(p-1)$ periods, the strategy fails. When needed to sample new probing spaces for depths $p' > p-1$, use those provided by the oracle in line 1 of Game 9.4 (this was already done for $p' > p$ before, now we also use these for depth $p' = p$).
3. Set Memory to be the memory of the algorithm, and the exploratory queries for depths $p' > p-1$ to be exactly as in the beginning of the it-th period of depth $p-1$.

The complete strategy for Game 9.2 at depth $p-1$ is described in Algorithm 9.10.

We now estimate the probability of success of this strategy for the Depth-$(p-1)$ Game 9.4. Notice that after the strategy submits an M-bit message and exploratory queries to the oracle, in lines 4-9 of the Depth-$(p-1)$ Game 9.4, the oracle proceeds to simulate the run of the it-th depth-$(p-1)$ period of Depth-p Game 9.4. Indeed, the responses obtained by alg in the run initialized by Algorithm 9.10 are stochastically equivalent to those that were obtained during that it-th depth-$(p-1)$ period because they were generated using the same process. As a result, the final run in lines 4-9 of the Depth-$(p-1)$ Game 9.4 is stochastically equivalent to the last step of the following procedure:

1. Run the complete depth-p strategy for Depth-p Game 9.4.
2. Sample $\mathsf{it} \sim \mathcal{U}([K_0])$ independently from the previous run.
3. If there were no $\mathsf{it}-1$ finished depth-$(p-1)$ periods in the previous run, strategy fails.
4. Otherwise, Re-run the it-th depth-$(p-1)$ period with the exact same randomness as in item 1, for at most $T_{max}^{(p-1)}$ iterations.

In the rest of the proof, we will prove success probabilities for this construction. Note that on $\mathcal{E}$, because K_0 depth-$(p-1)$ periods were completed, the strategy does not fail at step 3 and step 4 exactly implements the it-th depth-$(p-1)$ period of Depth-p Game 9.4. Further, this period will be complete, given enough iterations to be finished. That is, if it uses at most $T_{max}^{(p-1)}$ iterations, the player wins at the Depth-$(p-1)$ Game 9.4. We therefore

Input: dimensions d, $\tilde{d}$, number of vectors k, depth p, M-bit memory algorithm alg for Game 9.4 at depth p

Output: strategy for Game 9.4 at depth $p-1$

1 Receive subspaces $E_1, \dots, E_P$ and probing subspaces $V_j^{(p-1+i)}$ for $i \in [P-p+1]$ and $j \in [k]$

2 Initialize memory of alg with the same M-bit message $\mathsf{Message}$ and define $n_{p+i} \in [k]$ and exploratory vectors $\boldsymbol{y}_j^{(p+i)}$ for $i \in [n_{p+i}]$ for all $i \in [P-p]$ as in the Depth-p Game 9.4 (ignoring probing subspaces $V_j^{(p)}$)

3 Set $n_{p'} \leftarrow 0$ for $p' \in [p]$, reset probing subspaces $V_j^{(p+i)}$ for $i \in [P-p]$ and $j > n_{p+i}$

4 Sample $\mathsf{it} \sim \mathcal{U}([K_0])$. Initialize $\mathsf{EnoughPeriods} \leftarrow$**false**

5 **for** $t \in [T_{max}^{(p)}]$ **do**

6 Run alg with current memory to get query $\boldsymbol{x}_t$. Update exploratory queries and probing subspaces as in Procedure 9.3 and **return** $\boldsymbol{g}_t = \mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t)$ as response to alg. When needed to sample a depth-p probing subspace $V_j^{(p)}$, use one provided by the oracle in line 1 that was not yet used, and with smallest index j.

7 **if** n_{p-1} *was reset because of deeper periods* **then** Strategy fails, **end** procedure;

8 **if** n_{p-1} *was reset for* it*-th time (counting* $t = 1$*s)* **then** $\mathsf{EnoughPeriods} \leftarrow$**true**, $t(\mathsf{it}) \leftarrow t$;

9 **end**

10 **if** $\mathsf{EnoughPeriods}$ **then**

11 Submit to the oracle the memory state Memory of alg, as well as all values of n_{p-1+i} for $i \in [P-p+1]$ and exploratory queries $\boldsymbol{y}_j^{(p-1+i)}$ for $j \in [n_{p-1+i}], i \in [P-p+1]$, just before starting iteration $t(\mathsf{it})$

12 **else** Strategy fails, **end** procedure;

Algorithm 9.10: Strategy of the Player for Depth-$(p-1)$ Game 9.4 given a strategy for depth p

aim to bound the number of iterations $\mathsf{length}(i)$ needed for the i-th depth-$(p-1)$ period, with the convention $\mathsf{length}(i) := \infty$ if this period was never finished. We have

$$\mathbb{E}[\mathsf{length}(\mathsf{it}) \mid \mathcal{E}] = \frac{1}{K_0} \sum_{i=1}^{K_0} \mathsf{length}(i) \le \frac{T_{max}^{(p)}}{K_0}.$$

As a result, letting $\mathcal{F} = \left\{ \mathsf{length}(\mathsf{it}) \le T_{max}^{(p-1)} := \frac{8PT_{max}^{(p)}}{K_0} \right\}$, we have

$$\mathbb{P}(\mathcal{F} \mid \mathcal{E}) \ge 1 - \frac{1}{8P}.$$

In summary, on $\mathcal{E} \cap \mathcal{F}$, Algorithm 9.10 wins at Game 9.4 at depth $p-1$. Thus,

$$\mathbb{P}(\text{Algorithm 9.10 wins}) \ge \mathbb{P}(\mathcal{E} \cap \mathcal{F}) \ge \left(q - \frac{3}{8P}\right)\left(1 - \frac{1}{8P}\right) \ge q - \frac{1}{2P}.$$

Because $T_{max}^{(p-1)} \leq \frac{32Plk}{\tilde{d}} T_{max}^{(p)}$, this ends the proof of the result. ■

We now apply Lemma 9.7 recursively to progressively reduce the depth of Game 9.4. This gives the following result.

Theorem 9.6. *Let $P \geq 2$, $d \geq 40P$. Suppose that*

$$c_2 \frac{M + dP \ln d}{d} P^3 \ln d \leq k \leq c_1 \frac{d}{C_\alpha P \ln d} \tag{9.29}$$

for some universal constants $c_1, c_2 > 0$. If a strategy for Game 9.4 for depth P uses M bits of memory and wins with probability at least $\frac{1}{2}$, then it performed at least

$$T_{max} \geq \frac{k}{2} \left(\frac{d}{100P^2 lk} \right)^{P-1}$$

queries.

Proof Define for any $p \in [P]$,

$$T_{max}^{(P)} = \frac{k}{2} \left(\frac{\tilde{d}}{32Plk} \right)^{p-1}.$$

Suppose for now that the parameter k satisfies all assumptions from Lemma 9.6 for all $p \in \{2, \dots, P\}$. Then, starting from a strategy for Game 9.4 at depth P and that wins with probability $q \geq \frac{1}{2}$, Lemma 9.7 iteratively constructs strategies for $p \in [P]$ for Game 9.4 at depth p with $T_{max}^{(p)}$ iterations that wins with probability $q - \frac{1}{2P}(P-p)$. Now to win Game 9.4 with depth 1, one needs to make at least k queries (the exploratory queries). Hence, no algorithm wins with such probability $q - \frac{1}{2P}(P-1)$ using $T_{max}^{(1)}$ queries. Recall that $\tilde{d} \geq d/(3P)$, hence

$$T_{max}^{(P)} \geq \frac{k}{2} \left(\frac{d}{100P^2 lk} \right)^{P-1}.$$

The only remaining step is to check that all assumptions from Lemma 9.6 are satisfied. It suffices to check that

$$\frac{\tilde{d}}{4l} \geq k \geq 50 \cdot (4P) \frac{M + \frac{d}{8} \log_2(T_{max}^{(P)}) + 3}{\tilde{d}} \ln \frac{\sqrt{d}}{\delta_1}. \tag{9.30}$$

We start with the upper bound. Recalling the definition of l in Eq (9.7), we have that

$$\frac{\tilde{d}}{4l} \geq \frac{d}{12Pl} = \Omega\left(\frac{d}{C_\alpha P \ln d} \right).$$

Now for the upper bound,

$$50 \cdot (4P) \frac{M + \frac{d}{8} \log_2(T_{max}^{(P)}) + 3}{\tilde{d}} \ln \frac{\sqrt{d}}{\delta_1} = \mathcal{O}\left(\frac{M + dP \ln d}{d} P^3 \ln d \right).$$

Hence, for a choice of constants $0 < c_1 < c_2$ that we do not specify, Eq (9.30) holds. ■

9.4.6 Reduction from the feasibility procedure to the feasibility game

The only step remaining is to link Procedure 9.3 to the Game 9.4 at depth P. Precisely, we show that during a run of Procedure 9.3, many depth-P periods are completed – recall that Game 9.4 at depth P exactly corresponds to the run of a depth-P period of Procedure 9.3. For this, we first prove a simple query lower bound for the following game, that emulates the discovery of the subspace E_P at the last layer P.

Input: d, subspace dimension $\tilde{d}$, number of samples m

1 *Oracle:* Sample a uniformly random $\tilde{d}$-dimensional linear subspace E of $\mathbb{R}^d$

2 *Oracle:* Send i.i.d. samples $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \overset{i.i.d.}{\sim} \mathcal{U}(S_{d-1} \cap E)$ to player

3 *Player:* Based on $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ output a unit vector $\boldsymbol{y}$

4 The learner wins if $\|\mathrm{Proj}_E(\boldsymbol{y})\| < \sqrt{\frac{\tilde{d}}{20d}}$.

Game 9.11: Kernel discovery game

We show that to win the Kernel discovery Game 9.11 with reasonable probability, one needs $\Omega(\tilde{d})$ queries. This is to be expected since finding orthogonal vectors to E requires having information on the complete $\tilde{d}$-dimensional space.

Lemma 9.8. *Let $m \leq \frac{\tilde{d}}{2} \leq \frac{d}{4}$. No algorithm wins at Game 9.11 with probability more than $e^{-\tilde{d}/10}$.*

Proof Suppose that $m \leq \frac{\tilde{d}}{2}$. First note that with probability one, the samples $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ are all linearly independent. Conditional on $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$, the space E can be decomposed as

$$E = Span(\boldsymbol{v}_1, \dots, \boldsymbol{v}_m) \oplus F$$

where under $\mathcal{E}$, $F = E \cap Span(\boldsymbol{v}_i, i \in [m])^\perp$ is a uniform $(\tilde{d} - m)$-dimensional subspace of $Span(\boldsymbol{v}_i, i \in [m])^\perp$. Now letting $\boldsymbol{z} = \mathrm{Proj}_{Span(\boldsymbol{v}_i, i \in [m])^\perp}(\boldsymbol{y})$, one has

$$\|\boldsymbol{z}\|^2 = \|\boldsymbol{y}\|^2 - \|\mathrm{Proj}_{Span(\boldsymbol{v}_i, i \in [m])}(\boldsymbol{y})\|^2 = 1 - \|\boldsymbol{y} - \boldsymbol{z}\|^2. \tag{9.31}$$

Further, provided that $\boldsymbol{z} \neq 0$, from the point of view of F, the vector $\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$ is a random uniform unit vector in $Span(\boldsymbol{v}_1, \dots, \boldsymbol{v}_m)^\perp$. Formally, Lemma 9.14 shows that

$$\mathbb{P}\left(\|\mathrm{Proj}_F(\boldsymbol{z})\| < \|\boldsymbol{z}\| \sqrt{\frac{\tilde{d} - m}{10(d - m)}} \right) \leq e^{-(\tilde{d}-m)/5} \leq e^{-\tilde{d}/10}$$

In the last inequality, we used $m \leq \frac{\tilde{d}}{2}$. We denote by $\mathcal{F}$ the complement of this event. Then, under $\mathcal{E} \cap \mathcal{F}$,

$$\begin{aligned}
\|\mathrm{Proj}_E(\boldsymbol{y})\|^2 &= \|\boldsymbol{y} - \boldsymbol{z}\|^2 + \|\mathrm{Proj}_F(\boldsymbol{z})\|^2 \\
&\geq \|\boldsymbol{y} - \boldsymbol{z}\|^2 + \frac{\tilde{d} - m}{10(d - m)} \|\boldsymbol{z}\|^2 \\
&\geq \frac{\tilde{d} - m}{10(d - m)} + \|\boldsymbol{y} - \boldsymbol{z}\|^2 \left(1 - \frac{\tilde{d} - m}{10(d - m)} \right) \geq \frac{\tilde{d}}{20d}.
\end{aligned}$$

In the second inequality we used Eq (9.31). In summary, the player loses with probability $\mathbb{P}(\mathcal{E} \cap \mathcal{F}) \geq 1 - e^{-\tilde{d}/10}$. This ends the proof. ■

Using a reduction from Procedure 9.3 to the Kernel Discovery Game 9.11, we use the previous query lower bound to show that to solve Procedure 9.3, the algorithm needs to complete $\Omega(d/(Plk))$ depth-P periods.

Lemma 9.9. *Let alg be an algorithm for Procedure 9.3. Suppose that $4l_P k \leq \tilde{d}$ and $l_P \geq l$. Then, with probability at least $1 - \frac{1}{8P} - e^{-\tilde{d}/10}$, during the run of Procedure 9.3, there were are least $\frac{\tilde{d}}{2l_P k}$ completed periods of depth P.*

Proof Similarly as in the proof of Lemma 9.6, by Lemma 9.2, we know that on an event $\mathcal{E}$ of probability at least $1 - 4d^2 e^{-l_P/16}$, the first d periods of depth P are proper. In our context, this means that under $\mathcal{E}$, if the algorithm *alg* was successful for Procedure 9.3 within the first d depth-P periods, then the final query $\boldsymbol{x}_t$ for which $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}_t) = \textsf{Success}$ satisfied $\|\text{Proj}_{E_P}(\boldsymbol{x}_t)\| \leq \eta_P$. We can therefore construct a strategy for the Kernel Discovery Game 9.11 as follows: we simulate a run of Procedure 9.3 using $E_P = E$ the subspace provided by the oracle. When needed to construct a new depth-P probing subspace, we use l vectors of the sequence $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ similarly as in Algorithm 9.9 for the proof of Lemma 9.6. The strategy is formally defined in Algorithm 9.12.

Input: depth P, dimensions d, $\tilde{d}$, l, number of exploratory queries k, M-bit algorithm *alg*, number of samples $m = \left\lfloor \tilde{d}/2 \right\rfloor$

1 Sample independently $E_1, \ldots, E_{P-1}$, uniform $\tilde{d}$-dimensional subspaces of $\mathbb{R}^d$
2 Initialize $n_p \leftarrow 0$ for $p \in [P]$ and set memory of *alg* to $\mathbf{0}$
3 Receive samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ and set sample index $i \leftarrow 1$
4 **while** *alg did not receive a response* Success **do**
5 Run *alg* with current memory to obtain query $\boldsymbol{x}$. Update exploratory queries and probing subspaces as in lines 5-13 of Procedure 9.3. Whenever needed to sample a depth-P probing subspace $V_{n_P}^{(P)}$ of E_P:
6 **if** $i + l - 1 > m$ **then** Strategy fails: **end** procedure;
7 **else** use oracle samples, set $V_{n_P}^{(P)} := Span(\boldsymbol{v}_i, \ldots, \boldsymbol{v}_{i+l-1})$ and $i \leftarrow i + l$;
8 **return** $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x})$ as response to *alg*
9 **if** $\mathcal{O}_{\boldsymbol{V}}(\boldsymbol{x}) = \textsf{Success}$ **then return** $\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$ to oracle, **break**;
10 **end**

Algorithm 9.12: Strategy of the Player for the Kernel Discovery Game 9.11

From the previous discussion, letting $\mathcal{F}$ be the event that at most $n_0 := \left\lfloor \frac{\tilde{d}}{2l_P k} \right\rfloor$ depth-P periods were completed, we have that under $\mathcal{E} \cap \mathcal{F}$ Algorithm 9.12, needed at most $l_P k n_0 \leq \frac{\tilde{d}}{2}$ samples from the oracle and the last vector vector $\boldsymbol{x}$ satisfies $\|\text{Proj}_{E_P}(\boldsymbol{x})\| = \|\text{Proj}_E(\boldsymbol{x})\| \leq \eta_P$. Now recall that because $\boldsymbol{x}$ was successful for $\mathcal{O}_{\boldsymbol{V}}$ we must have $\boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2}$ and as a

result $\|\boldsymbol{x}\| \geq \frac{1}{2}$. Hence, the output vector $\boldsymbol{y} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$ satisfies

$$\|\mathrm{Proj}_E(\boldsymbol{y})\| \leq 2\eta_P < \sqrt{\frac{\tilde{d}}{20d}}.$$

In the last inequality, we used Eq (9.8). Hence, using Lemma 9.8, we have

$$\mathbb{P}(\mathcal{E} \cap \mathcal{F}) \leq \mathbb{P}(\textit{Algorithm 9.12 wins}) \leq e^{-\tilde{d}/10}.$$

In particular,

$$\mathbb{P}(\mathcal{F}) \leq \mathbb{P}(\mathcal{E}^c) + \mathbb{P}(\mathcal{E} \cap \mathcal{F}) \leq 4d^2 e^{-l_P/16} + e^{-\tilde{d}/10} \leq \frac{1}{8P} + e^{-\tilde{d}/10}.$$

In the last inequality we used $l_P \geq l \geq 16\ln(32d^2P)$ from Eq (9.7). Because on $\mathcal{F}^c$ there are at last $n_0 + 1 \geq \frac{\tilde{d}}{2lk}$ completed depth-P periods, this ends the proof. ■

Lemma 9.9 shows that many depth-P periods are performed during a run of Procedure 9.3. Because the depth-P Game 9.4 exactly simulates a depth-P period of Procedure 9.3, we can combine this with our previous lower bound to obtain the following.

Theorem 9.7. *Let $P \geq 2$ and $d \geq 20P$. Suppose that k satisfies Eq (9.29) as in Theorem 9.6. Also suppose that $4l_pk \leq \tilde{d}$ and $l_P \geq l$. If an algorithm for Procedure 9.3 uses M bits of memory and wins making at most T_{max} queries with probability at least $\frac{3}{4}$, then*

$$T_{max} \geq \frac{kl}{l_P}\left(\frac{d}{100P^2lk}\right)^P.$$

Proof Lemma 9.9 plays exactly the same role as Lemma 9.6. In fact, we can easily check that the exact same proof as for Lemma 9.7 shows that if there is an algorithm for Procedure 9.3 that uses at most T_{max} queries and wins with probability at least $q \geq \frac{3}{4}$, then there is a strategy for Game 9.4 for depth P, that uses the same memory and at most

$$T_{max}^{(P)} := \frac{8P}{\tilde{K}_0} T_{max}$$

queries, where $\tilde{K}_0 = \frac{\tilde{d}}{2l_Pk}$ is the number of depth-P periods guaranteed by Lemma 9.9. Further, it wins with probability at least $q - \frac{1}{8P} - e^{-\tilde{d}/10}$. The failure probability $\frac{1}{8P} + e^{-\tilde{d}/10}$ corresponds to the failure probability of Lemma 9.9. Hence this win probability is more than $\frac{1}{2}$ since $P \geq 2$ and $\tilde{d} \geq \frac{d}{2} \geq 20$. By Theorem 9.6 we must have

$$T_{max} = \frac{\tilde{d}}{16Pl_Pk} T_{max}^{(P)} \geq \frac{\tilde{d}}{16Pl_Pk}\frac{k}{2}\left(\frac{d}{100P^2lk}\right)^{P-1} \geq \frac{kl}{l_P}\left(\frac{d}{100P^2lk}\right)^P.$$

This ends the proof. ■

From Lemma 9.1, we know that with high probability on $E_1, \ldots, E_P$, Procedure 9.3 implements a valid feasibility problem for accuracy $\epsilon = \delta_1/2$. Combining this with the previous query lower bound for Procedure 9.3 gives the desired final result for deterministic algorithms.

Theorem 9.8. *Fix $\alpha \in (0,1]$, $d \geq 1$ and an accuracy $\epsilon \in (0, \frac{1}{\sqrt{d}}]$ such that*

$$d \ln^2 d \geq \left(\frac{c_3}{\alpha}\right)^{\frac{\ln 2}{\alpha}} \cdot \frac{M}{d} \ln^4 \frac{1}{\epsilon}$$

for some universal constant c_3. Then, any M-bit deterministic algorithm that solves feasibility problems up to accuracy ϵ makes at least

$$\left(\frac{d}{M}\right)^{\alpha} \frac{1}{\epsilon^{2\psi(d,M,\epsilon)}}$$

queries, where $\psi(d,M,\epsilon) = \frac{1-\ln\left(\frac{M}{d}\right)/\ln d}{1+(1+\alpha)\ln\left(\frac{M}{d}\right)/\ln d} - \mathcal{O}\left(\frac{\ln \frac{\ln(1/\epsilon)}{\ln d}}{\ln d} + \frac{\ln\ln d}{\ln d}\right)$.

Proof Roughly speaking, the proof consists in finding parameters for P and k that (1) satisfy the assumptions to apply the query lower bound Theorem 9.7 on Procedure 9.3, (2) for which this procedure emulates ϵ-accuracy feasibility problems with high accuracy, and (3) that maximizes the query lower bound provided in Theorem 9.7.

First, we recall that $d \log_2 \frac{1}{2\epsilon}$ bits of memory are necessary to solve the feasibility problem because this is already true for optimizing 1-Lipschitz functions on the unit ball [WS19, Theorem 5]. Without loss of generality, we therefore suppose that $M \geq d \ln \frac{1}{\epsilon}$. For convenience, let us define the following quantity

$$\tilde{P} := \frac{2\ln\frac{1}{\epsilon} - (1+\alpha)\ln\frac{M}{d} - (4+3\alpha)\ln\left(\frac{\ln(1/\epsilon)}{\ln d}+1\right) - 2\ln(c_2 \ln d) - 24}{\ln d + (1+\alpha)\ln\frac{M}{d} + 3(1+\alpha)\ln\left(\frac{\ln(1/\epsilon)}{\ln d}+1\right) + \ln(c_2 \ln d) + 12}.$$

We now define the parameters P, k, and l_P as

$$P := \left\lceil \tilde{P} \right\rceil, \quad k := \left\lceil 3c_2 \frac{M}{d} P^3 \ln d \right\rceil, \quad \text{and} \quad l_P := l \vee \left\lfloor \left(\frac{\tilde{d}}{4k}\right)^{P-\tilde{P}} \right\rfloor.$$

In particular, note that $P < P_{max} := \frac{2\ln\frac{1}{\epsilon}}{\ln d} + 1$ and we directly have $l_P \geq l$. We also recall that under the assumptions from Theorem 9.7, we showed that $4lk \leq \tilde{d}$. As a result, we also have $4l_P k \leq \tilde{d}$ provided that Eq (9.29) is satisfied. Now fix an algorithm alg for the ϵ-accuracy feasibility problem that uses at most M bits of memory and uses at most T_{max} separation oracle queries. By assumption, we have

$$d \ln d \geq \sqrt{c_3} \ln \frac{1}{\epsilon}.$$

Hence, setting $c_3 \geq 20^2$ we have that $P \leq P_{max} \leq \frac{d}{20}$. Assuming now that $P \geq 2$, we can apply Lemma 9.1 which shows that on an event $\mathcal{E}$ of probability at least $1 - e^{-d/40} \geq 1 - e^{-2} > \frac{3}{4}$, Procedure 9.3 using alg emulates a valid $(\delta_1/2)$-accuracy feasibility problem. Note that

$$\frac{\delta_1}{2} = \frac{1}{720\mu^{P-2}\mu_P}\sqrt{\frac{l}{dk^{\alpha}}} = \frac{5}{6\mu^P}\sqrt{\frac{kl_P}{l}} \geq \frac{1}{\mu^{\tilde{P}}}\sqrt{\frac{k}{2l}\left(\frac{\tilde{d}}{4k\mu^2}\right)^{P-\tilde{P}}}.$$

In the inequality, we used $k \geq 3$. Furthering the bounds and using $P - \tilde{P} < 1$, we obtain

$$\frac{\delta_1}{2} \geq \frac{1}{\mu^{\tilde{P}}} \sqrt{\frac{\tilde{d}}{8l\mu^2}} \geq \frac{1}{3000\mu^{\tilde{P}}\sqrt{Pk^{1+\alpha}}} \geq \epsilon.$$

In the last inequality, we used the definition of $\tilde{P}$, which using $\alpha \leq 1$, implies in particular

$$\tilde{P} \leq \frac{2\ln\frac{1}{\epsilon} - \ln P_{max} - (1+\alpha)\ln(6c_2\frac{M}{d}P_{max}^3 \ln d) - 2\ln(3000)}{\ln d + (1+\alpha)\ln\left(6c_2\frac{M}{d}P_{max}^3 \ln d\right) - \ln(16\ln d) + 2\ln(600)} \leq \frac{\ln\frac{1}{\epsilon} - \ln(3000\sqrt{Pk^{1+\alpha}})}{\ln\mu}.$$

As a result, because $\delta_1/2 \geq \epsilon$, under that same event $\mathcal{E}$, Procedure 9.3 terminates with at most T_{max} queries. We now check that the choice of k satisfies Eq (9.29). Note that $dP\ln d \leq 2d\ln\frac{1}{\epsilon}$. As a result, $M + dP\ln d \leq 3M$ and hence k directly satisfies the left-hand side inequality of Eq (9.29). The assumption gives

$$d\ln^2 d \geq \frac{100C_\alpha c_2}{c_1}\frac{M}{d}\ln^4\frac{1}{\epsilon}.$$

As a result, $c_1\frac{d}{C_\alpha P\ln d} \geq c_1\frac{d}{C_\alpha P_{max}\ln d} \geq 6c_2(M/d)P_{max}^3\ln d \geq k$. As a result, the right-hand side is of Eq (9.29) is also satisfied. We can now apply Theorem 9.7 which gives

$$\begin{aligned} T_{max} \geq \frac{kl}{l_P}\left(\frac{d}{100P^2lk}\right)^P \geq \frac{k}{(13Pl)^{P-\tilde{P}}}\left(\frac{d}{100P^2lk}\right)^{\tilde{P}} &\geq \frac{c_2P^2(M/d)\ln d}{5l}\left(\frac{d}{100P^2lk}\right)^{\tilde{P}} \\ &=: \frac{1}{\epsilon^{2\tilde{\psi}(d,M,\epsilon)}}, \end{aligned}$$

where we defined $\tilde{\psi}(d,M,\epsilon)$ through the last equality. Note that the above equation always holds, even if $P < 2$ (that is, $\tilde{P} \leq 1$) because in that case it is implied by $T_{max} \geq d$ (which is necessary even for convex optimization). We wrote the above equation for the sake of completeness; the above computations can be simplified to

$$\begin{aligned} \tilde{\psi}(d,M,\epsilon) &= \frac{\ln\frac{M}{d}}{2\ln\frac{1}{\epsilon}} + \mathcal{O}\left(\frac{\ln\ln\frac{1}{\epsilon}}{\ln\frac{1}{\epsilon}}\right) + \frac{\tilde{P}\ln\frac{d}{100P^2lk}}{2\ln\frac{1}{\epsilon}} \\ &= \frac{\ln\frac{M}{d}}{2\ln\frac{1}{\epsilon}} + \frac{\ln d - \ln\frac{M}{d} - \frac{1+\alpha}{2}\frac{\ln d\cdot\ln\frac{M}{d}}{\ln\frac{1}{\epsilon}} - 4\ln\frac{\ln(1/\epsilon)}{\ln d}}{\ln d + (1+\alpha)\ln\frac{M}{d} + 3(1+\alpha)\ln\frac{\ln(1/\epsilon)}{\ln d}} + \mathcal{O}\left(\frac{\ln\ln d}{\ln d}\right) \\ &\geq -\frac{\alpha\ln\frac{M}{d}}{2\ln\frac{1}{\epsilon}} + \frac{\ln d - \ln\frac{M}{d} - 4\ln\frac{\ln(1/\epsilon)}{\ln d}}{\ln d + (1+\alpha)\ln\frac{M}{d} + 3(1+\alpha)\ln\frac{\ln(1/\epsilon)}{\ln d}} + \mathcal{O}\left(\frac{\ln\ln d}{\ln d}\right). \end{aligned}$$

This ends the proof of the theorem. ■

In the standard regime when $\ln\frac{1}{\epsilon} \leq d^{o(1)}$, the query lower bound from Theorem 9.8 can be greatly simplified, and directly implies Theorem 9.1.

9.5 Query Complexity / Memory Trade-offs for Randomized Algorithms

The feasibility procedure defined in Procedure 9.3 is adaptive in the algorithm queries. As a result, this approach fails to give query lower bounds for randomized algorithms, which is the focus of the present section.

Although Procedure 9.3 is adaptive, note that the generated subspaces $E_1, \dots, E_P$ and probing subspaces $V_i^{(p)}$ for $i \in [k]$ and $p \in [P]$ are not. In fact, the only source of adaptivity comes from deciding when to add a new probing subspace for any depth $p \in [P]$. In Procedure 9.3 this is done when the algorithm performs a depth-p exploratory query. We now present an alternative feasibility procedure for which the procedure oracle does not need to know when exploratory queries are performed, at the expense of having worse query lower bounds.

9.5.1 Definition of the hard class of feasibility problems

The subspaces $E_1, \dots, E_P$ are sampled exactly as in Procedure 9.3 as independent uniform $\tilde{d}$-dimensional subspaces of $\mathbb{R}^d$ where $\tilde{d} = \lfloor d/(2P) \rfloor$. As before, for each space E_p for $p \in [P]$ we construct l-dimensional probing subspaces. However, given such probing subspaces say $V_1, \dots, V_r$ for $r \in [k]$, we define a different depth-p oracle. We will always assume that $k \geq 3$. Precisely, given parameters $\boldsymbol{\delta} = (\delta_1, \dots, \delta_k) \in (0, \infty)^k$, we first define the set

$$\mathcal{I}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}) = \left\{ i \in [k] : \|\mathrm{Proj}_{V_i}(\boldsymbol{x})\| > \delta_i \right\}.$$

The oracle is then defined as

$$\tilde{\boldsymbol{g}}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}) := \begin{cases} \frac{\mathrm{Proj}_{V_i}(\boldsymbol{x})}{\|\mathrm{Proj}_{V_i}(\boldsymbol{x})\|} & \text{if } \mathcal{I}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}) \neq \emptyset \text{ and } i = \min \mathcal{I}_{V_1,\dots,V_r}(\boldsymbol{x}; \boldsymbol{\delta}), \\ \mathsf{Success} & \text{otherwise.} \end{cases} \tag{9.32}$$

Compared to $\boldsymbol{g}_{V_1,\dots,V_r}$, this oracle does not combine probing subspaces by taking their span, and prioritizes separation hyperplanes constructed from probing subspaces with the smallest index $i \in [k]$. In the oracle, only functions $\tilde{\boldsymbol{g}}_{V_1,\dots,V_k}$ which have exactly k subspaces are used – the definition for $r < k$ subspaces will only be useful for the proof.

For each depth $p \in [P]$, we will sample k probing subspaces $V_1^{(p)}, \dots, V_k^{(p)}$ as before: these are i.i.d. l_p-dimensional subspaces of E_p. This time, we set

$$l_p = l := \left\lceil Ck^3 \ln d \right\rceil, \quad p \in [P-1], \tag{9.33}$$

for a universal constant $C \geq 1$ introduced in Lemma 9.10. We let $l_P \in [\tilde{d}]$ with $l_P \geq l$ be a parameter as in the deterministic case. Also, these probing subspaces will be resampled regularly throughout the feasibility procedure. We use the notation $\boldsymbol{V}^{(p)} = (V_1^{(p)}, \dots, V_k^{(p)})$, noting that here $\boldsymbol{V}^{(p)}$ always contains all k probing subspaces contrary to Procedure 9.3.

Given these subspaces, the format of the oracle is similar as in Eq (9.6):

$$\tilde{\mathcal{O}}_{\boldsymbol{V}^{(1)},\dots,\boldsymbol{V}^{(P)}}(\boldsymbol{x}) := \begin{cases} \boldsymbol{e} & \text{if } \boldsymbol{e}^\top \boldsymbol{x} > -\frac{1}{2} \\ \tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \boldsymbol{\delta}^{(p)}) & \text{if } \tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}; \boldsymbol{\delta}^{(p)}) \neq \mathsf{Success} \\ & \text{and } p = \min\left\{q \in [P],\ \tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(q)}}(\boldsymbol{x}; \boldsymbol{\delta}^{(q)}) \neq \mathsf{Success}\right\} \\ \mathsf{Success} & \text{otherwise.} \end{cases} \tag{9.34}$$

Before defining the parameters $\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(P)}$, we first define $\eta_1, \dots, \eta_P$ as follows. First, let

$$\eta_P := \frac{1}{10}\sqrt{\frac{\tilde{d}}{d}} \quad \text{and} \quad \mu_P := 1200k\sqrt{\frac{kd}{l_P}},$$

and for all $p \in [P-1]$ we let

$$\eta_p := \frac{\eta_P/\mu_P}{\mu^{P-p-1}} \quad \text{where} \quad \mu := 1200k\sqrt{\frac{kd}{l}} \tag{9.35}$$

We then define the orthogonality tolerance parameters as follows

$$\delta_i^{(p)} := \frac{\eta_p(1-2/k)^{k-i}}{2}\sqrt{\frac{l_p}{\tilde{d}}}, \quad p \in [P], i \in [k]. \tag{9.36}$$

As for the deterministic case, whenever the output of the oracle is not $\boldsymbol{e}$ nor $\mathsf{Success}$, we say that $\boldsymbol{x}$ is a depth-p query where $p = \min\left\{q \in [P], \tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(q)}}(\boldsymbol{x}; \boldsymbol{\delta}^{(q)}) \neq \mathsf{Success}\right\}$. Except for the probing subspaces, all other parameters are kept constant throughout the feasibility problem. The depth-p probing subspaces are resampled independently from the past every T_p iterations, where the sequence $T_1, \dots, T_P$ is defined as

$$T_p := \left\lfloor \frac{k}{2} \right\rfloor N^{p-1}, \quad p \in [P], \quad \text{where} \quad N := \left\lfloor \frac{\tilde{d}}{2lk} \right\rfloor.$$

We are now ready to formally define our specific separation oracle for randomized algorithms. As in the oracle of Procedure 9.3, we use the fallback separation oracle $\mathcal{O}_{E_1,\dots,E_P}$ when the one from Eq (9.34) returns $\mathsf{Success}$. Having sampled i.i.d. $\tilde{d}$-dimensional subspaces $E_1, \dots, E_P$, we independently construct for each $p \in [P]$ an i.i.d. sequence $(\boldsymbol{V}^{(p,a)})_{a \geq 0}$ of lists $\boldsymbol{V}^{(p,a)} = (V_1^{(p,a)}, \dots, V_k^{(p,a)})$ containing i.i.d. uniform l_p-dimensional random subspaces of E_p. To make the notations cleaner we assume that the number of iterations starts from $t = 0$. We define the separation oracle for all $t \geq 0$ and $\boldsymbol{x} \in \mathbb{R}^d$ via

$$\tilde{\mathcal{O}}_t(\boldsymbol{x}) := \begin{cases} \tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}(\boldsymbol{x}) & \text{if } \tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}(\boldsymbol{x}) \neq \mathsf{Success} \\ \mathcal{O}_{E_1,\dots,E_P}(\boldsymbol{x}) & \text{otherwise.} \end{cases} \tag{9.37}$$

This definition is stochastically equivalent to simply resampling the depth-p probing subspaces $\boldsymbol{V}^{(p)}$ every T_p iterations. We take this perspective from now on. In this context, a

depth-p period is simply a interval of time of the form $[aT_p, (a+1)T_p)$ for some integer $a \geq 0$. Here, the feasible set is defined as

$$\tilde{Q}_{E_1,\dots,E_P} := B_d(\mathbf{0},1) \cap \left\{ \boldsymbol{x} : \boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2} \right\} \cap \bigcap_{p\in[P]} \left\{ \boldsymbol{x} : \|\mathrm{Proj}_{E_p}(\boldsymbol{x})\| \leq \delta_1^{(p)} \right\}.$$

We now prove query lower bounds for algorithms under this separation oracle using a similar methodology as for Procedure 9.3. To do so, we first need to slightly adjust the notion of exploratory queries in this context. At the beginning of each depth-p period, these are reset and we consider that there are no exploratory queries.

Definition 9.4 (Exploratory queries, randomized case)**.** *Let $p \in [P]$ and fix a depth-p period $[aT_p, (a+1)T_p)$ for $a \in \{0, \dots, N^{P-p} - 1\}$. Given previous exploratory queries $\boldsymbol{y}_1^{(p)}, \dots, \boldsymbol{y}_{n_p}^{(p)}$ in this period, we say that $\boldsymbol{x} \in B_d(\mathbf{0}, 1)$ is a depth-p exploratory query if*

1. *$\boldsymbol{e}^\top \boldsymbol{x} \leq -\frac{1}{2}$,*
2. *the query passed all probes from levels $q < p$, that is $\boldsymbol{g}_{\boldsymbol{V}^{(q)}}(\boldsymbol{x}; \boldsymbol{\delta}^{(q)}) = \mathsf{Success}$,*
3. *and it is robustly-independent from all previous depth-p exploratory queries in the period*

$$\|Proj_{Span(\boldsymbol{y}_r^{(p)}, r \leq n_p)^\perp}(\boldsymbol{x})\| \geq \gamma_p := \frac{\delta_1^{(p)}}{4k}. \tag{9.38}$$

We now introduce the feasibility game associated to the oracle which differs from the feasibility problem with the oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ in the following ways.

- The player can access some initial memory about the subspaces $E_1, \dots, E_P$.
- Their goal is to perform k depth-P exploratory queries during a single depth-P period of T_P iterations. In particular, they only have a budget of T_P calls to the oracle.
- They have a mild influence on the sequences of probing subspaces $\mathbb{V}^{(p)} := (\boldsymbol{V}^{(p,a)})_{a\geq 0}$ for $p \in [P]$ that will be used by the oracle. Precisely, for each $p \in [P]$, the oracle independently samples J_P i.i.d. copies of these sequences $\mathbb{V}^{(p,1)}, \dots, \mathbb{V}^{(p,J_P)}$ for a pre-specified constant J_P. The player then decides on an index $\hat{j} \in [J_P]$ and the oracle uses the sequences $\mathbb{V}^{(p,\hat{j})}$ for $p \in [P]$ to simulate the feasibility problem.

The details of the game are given in Game 9.13. Note that compared to the feasibility Game 9.4, we do not need to introduce specific games at depth $p \in [P]$ nor introduce exploratory queries. The reason is that because they are non-adaptive, the oracles at depth $p' > p$ do not provide any information about E_p. Hence, the game at depth $p \in [P]$ can simply be taken as the original game but with p layers instead of P.

Roughly speaking, the first step of the proof is to show that because of the construction of the oracle $\tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(p)}}$ for $p \in [P]$ in Eq (9.32), we have the following structure during any depth-p period with high probability. (1) The algorithm observes $V_1^{(p)}, \dots, V_k^{(p)}$ in this exact order and further, (2) the algorithm needs to query a new robustly independent vector to observe a new probing subspace. These are exactly the properties needed to replace the

Input: depth P; dimensions d, $\tilde{d}$, $l_1, \dots, l_P$; k; M-bit memory randomized algorithm alg; resampling horizons $T_1, \dots, T_P$; maximum index J_P

Oracle: Sample independently $E_1, \dots, E_P$, uniform $\tilde{d}$-dimensional subspaces of $\mathbb{R}^d$

Oracle: For all $p \in [P]$ sample independently J_P i.i.d. sequences $\mathbb{V}^{(p,1)}, \dots, \mathbb{V}^{(p,J_P)}$. Each sequence $\mathbb{V}^{(p,j)} = (\boldsymbol{V}^{(p,j,a)})_{a \in [0, N^{P-p})}$ contains N^{P-p} i.i.d. k-tuples $\boldsymbol{V}^{(p,j,a)} = (V_1^{(p,j,a)}, \dots, V_k^{(p,j,a)})$ of i.i.d. l_p-dimensional subspaces of E_p.

Player: Observe $E_1, \dots, E_P$ and all sequences $\mathbb{V}^{(p,1)}, \dots, \mathbb{V}^{(p,J_P)}$ for $p \in [P]$. Based on these, submit to oracle an M-bit message Message and an index $\hat{j} \in [J_P]$

Oracle: Initialize memory of alg to Message

Oracle: **for** $t \in \{0, \dots, T_P - 1\}$ **do**

  Run alg with current memory to get query $\boldsymbol{x}_t$

  if $t = 0 \bmod T_p$ **for** any $p \in [P]$ **then** $\boldsymbol{V}^{(p)} \leftarrow \boldsymbol{V}^{(p,\hat{j},t/T_p)}$

  return $\boldsymbol{g}_t = \tilde{\mathcal{O}}_{\boldsymbol{V}^{(1)}, \dots, \boldsymbol{V}^{(P)}}(\boldsymbol{x}_t)$ as response to alg

end

Player wins if the player performed k (or more) depth-P exploratory queries

Game 9.13: Feasibility Game for randomized algorithms

step from Procedure 9.3 in which the adaptive oracle adapts to when exploratory queries are performed. Proving this property is one of the main technicality to extend our query lower bounds for deterministic algorithms to randomized algorithms.

Lemma 9.10. *Fix $p \in [P]$ and $a \in \{0, \dots, N^{P-p} - 1\}$. Suppose that $l_p \geq Ck^3 \ln d$ for some universal constant $C > 0$ and $k \geq 3$. Then, with probability at least $1 - k^2 J_P e^{-k \ln d}$, for all times $t \in [aT_p, (a+1)T_p)$ during Game 9.13, the following hold. Let $r_p(t)$ be the number of depth-p exploratory queries in $[aT_p, t]$. If $r_p(t) \leq k$,*

- *the response $\boldsymbol{g}_t$ is consistent to the oracle without probing subspaces $V_i^{(p)}$ for $i > r_p(t)$, that is, replacing $\tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(p)}}(\cdot; \boldsymbol{\delta}^{(p)})$ with $\tilde{\boldsymbol{g}}_{V_1^{(p)}, \dots, V_{r_p(t)}^{(p)}}(\cdot; \boldsymbol{\delta}^{(p)})$ in Eq (9.34),*
- *if $\boldsymbol{x}_t$ is a depth-p' query for $p' > p$, then $\|Proj_{E_p}(\boldsymbol{x}_t)\| \leq \eta_p$.*

Here, the first bullet point exactly proves the behavior that was described above for the sequential discovery of probing subspaces. Roughly speaking, the second bullet point shows that periods are still mostly proper, at least before k exploratory queries are performed during the period.

Proof of Lemma 9.10 Fix $p \in [P]$ and the period index $a < N^{P-p}$. We will use the union bound to take care of the degree of liberty $\hat{j} \in [J_P]$. For now, fix $j \in [J_P]$ and suppose that we had $\hat{j} = j$, that is, all probing subspaces were constructed from the sequences $\mathbb{V}^{(1,j)}, \dots, \mathbb{V}^{(P,j)}$.

Let $i \in \{1, \dots, k\}$ and consider the game for which the oracle responses are constructed exactly as in Game 9.13 except during the considered period $[aT_p, (a+1)T_p)$ as follows. For the oracle response at time $t \in [aT_p, (a+1)T_p)$: if $r_p(t) \geq i$ we use the same oracle as in Game 9.13; but if $r_p(t) < i$, we replace $\boldsymbol{V}^{(p)}$ with $(V_1^{(p)}, \dots, V_{r_p(t)}^{(p)})$ – that is we ignore all

depth-p probing subspaces $V_j^{(p)}$ with index $j > r_p(t)$. For convenience, let us refer to this as Game$(p, a; i)$. Note that Game$(p, a; 1)$ is exactly Game 9.13. Indeed, at a given time t in the period, if there were no previous depth-p exploratory queries either (1) $\boldsymbol{x}_t$ does not pass probes at level $q < p$ or $\boldsymbol{e}^\top \boldsymbol{x} > -\frac{1}{2}$: in this case no depth-$p$ probing subspaces are needed to construct the response; or (2) we have in particular $\|\boldsymbol{x}\| \geq \frac{1}{2} \geq \gamma_p$ so $\boldsymbol{x}_t$ is exploratory. As a result, before the first depth-p exploratory query, having access to the depth-p subspaces is irrelevant.

Now fix $i \in [k-1]$. Our goal is to show that with high probability, the responses returned by Game$(p, a; i)$ are equivalent to those of Game$(p, a; i+1)$. Let $\mathcal{E}_i$ be the event that there were at least i depth-p exploratory queries during the period $[aT_p, (a+1)T_p)$. We recall the notations $\boldsymbol{y}_1^{(p)}, \boldsymbol{y}_2^{(p)}, \ldots$ for these exploratory queries. Note that we slightly abuse of notations because these are the exploratory queries for Game$(p, a; i)$ (not for Game 9.13). However, our goal is to show that their responses coincide so under this event, all these exploratory queries will coincide. Importantly, by construction of the oracle for Game$(p, a; i)$, all queries $\boldsymbol{y}_1^{(p)}, \ldots, \boldsymbol{y}_i^{(p)}$ are independent from the subspaces $V_s^{(p)} = V_s^{(p,j,a)}$ for all $s \geq i$. Indeed, during the period $[aT_p, (a+1)T_{p+1})$, before receiving the response for the query $\boldsymbol{y}_i^{(p)}$, the oracle only used probing subspaces $V_1^{(p)}, \ldots, V_{i-1}^{(p)}$. Hence, $\boldsymbol{y}_1^{(p)}, \ldots, \boldsymbol{y}_i^{(p)}$ are dependent only on those probing subspaces. As a result, from the point of view of the spaces $V_j^{(p)}$ for $j \geq i$, the subspace $F_i := Span(\mathrm{Proj}_{E_p}(\boldsymbol{y}_s^{(p)}), s \in [i])$ is uniformly random. Formally, we define the following event

$$\mathcal{F}_i = \bigcap_{j=i}^{k} \left\{ \left(1 - \frac{1}{2k}\right) \|\boldsymbol{x}\| \leq \sqrt{\frac{\tilde{d}}{l_p}} \|\mathrm{Proj}_{V_j}(\boldsymbol{x})\| \leq \left(1 + \frac{1}{2k}\right) \|\boldsymbol{x}\|, \ \forall \boldsymbol{x} \in F_i \right\}.$$

By Lemma 9.15 and the union bound, we have

$$\mathbb{P}(\mathcal{F}_i \mid \mathcal{E}_i) \geq 1 - k \exp\left(i \ln \frac{2C\tilde{d}k}{l_p} - \frac{l_p}{2^7 k^2} \right) \geq 1 - k \exp\left(k \ln d - \frac{l_p}{2^7 k^2} \right) \geq 1 - k e^{-k \ln d}.$$

Here we used $l_p \geq 2^8 C k^2 \ln d$ and the fact that $\dim(F_i) \leq i \leq k$. For convenience let $\zeta = \frac{1-1/(2k)}{1+1/(2k)}$. Using the previous bounds, under $\mathcal{E}_i \cap \mathcal{F}_i$, we have for any $\boldsymbol{y} \in Span(\boldsymbol{y}_s^{(p)}, p \in [i])$,

$$\|\mathrm{Proj}_{V_i}(\boldsymbol{y})\| \geq \zeta \|\mathrm{Proj}_{V_j}(\boldsymbol{y})\|, \quad j \in \{i+1, \ldots, k\}.$$

Now consider any time during the period $t \in [aT_p, (a+1)T_{p+1})$ such that $n_p(t) = i$. If $\boldsymbol{x}_t$ was not a depth-p' query with $p' \geq p$, knowing the depth-p probing subspaces is irrelevant to construct the oracle response. Otherwise, we have

$$\|\mathrm{Proj}_{Span(\boldsymbol{y}_s^{(p)}, s \in [i])^\perp}(\boldsymbol{x}_t)\| < \gamma_p. \tag{9.39}$$

Indeed, either $\boldsymbol{x}_t$ was a depth-p exploratory query and hence $\boldsymbol{x}_t = \boldsymbol{y}_i^{(p)}$, or it was not, in which case the third property from Definition 9.4 must not be satisfied (because the two first are already true since $\boldsymbol{x}_t$ is a depth-p' query with $p' \geq p$). For simplicity, write

$\boldsymbol{y}_t = \mathrm{Proj}_{Span(\boldsymbol{y}_s^{(p)}, s\in[i])}(\boldsymbol{x}_t)$ and note that $\|\boldsymbol{x}_t - \boldsymbol{y}_t\| < \gamma_p$ by Eq (9.39). Then, for any $i < j \leq k$,

$$\|\mathrm{Proj}_{V_i}(\boldsymbol{x}_t)\| > \|\mathrm{Proj}_{V_i}(\boldsymbol{y}_t)\| - \gamma_p \geq \zeta\|\mathrm{Proj}_{V_j}(\boldsymbol{x}_t)\| - \gamma_p > \zeta\|\mathrm{Proj}_{V_j}(\boldsymbol{x}_t)\| - (1+\zeta)\gamma_p.$$

In particular, if one has $j \in \mathcal{I}_{V_1,\dots,V_k}(\boldsymbol{x}_t; \boldsymbol{\delta}^{(i)})$ for some $i+1 < j \leq k$, we have

$$\|\mathrm{Proj}_{V_i}(\boldsymbol{x}_t)\| > \zeta\delta_j^{(p)} - 2\gamma_p \geq \zeta\delta_{i+1}^{(p)} - \frac{\delta_1^{(p)}}{2k} \geq \delta_i^{(p)}\left(\frac{\zeta}{1-2/k} - \frac{1}{2k}\right) \geq \delta_i^{(p)}.$$

In the second inequality, we use the definition of γ_p in Eq (9.38) and in the last inequality, we used $k \geq 3$. As a result, we also have $i \in \mathcal{I}_{V_1,\dots,V_k}(\boldsymbol{x}_t; \boldsymbol{\delta}^{(i)})$. Going back to the definition of the depth-p oracle in Eq (9.32) shows that in all cases (whether there is some $j \in \mathcal{I}_{V_1,\dots,V_k}(\boldsymbol{x}_t; \boldsymbol{\delta}^{(i)})$ for $i+1 < j \leq k$ or not),

$$\tilde{\boldsymbol{g}}_{\boldsymbol{V}^{(p)}}(\boldsymbol{x}_t; \boldsymbol{\delta}^{(p)}) = \tilde{\boldsymbol{g}}_{V_1^{(p)},\dots,V_i^{(p)}}(\boldsymbol{x}_t; \boldsymbol{\delta}^{(p)}),$$

that is, the oracle does not use the subspaces $V_j^{(p)}$ for $j > i$. In summary, under $\mathcal{E}_i \cap \mathcal{F}_i$ the responses provided in Game$(p, a; i)$ and Game$(p, a; i+1)$ are identical.

Using the above result recursively shows that under

$$\mathcal{G} := \bigcap_{i\in[k-1]} \mathcal{E}_i^c \cup (\mathcal{E}_i \cap \mathcal{F}_i),$$

the responses in Game$(p, a; k)$ are identical to those in Game$(p, a; 1)$ which is the original Game 9.13 provided $\hat{j} = j$. Hence, under $\mathcal{G}$ and assuming $\hat{j} = j$, the first claim of the lemma holds. The second point is a direct consequence of the previous property. For any fixed $t \in [aT_p, (a+1)T_{p+1})$ write $i = n_p(t)$. Under $\mathcal{G}$, because the event $\mathcal{E}_{n_p(t)} \cap \mathcal{F}_{n_p(t)}$ holds, we have in particular

$$\sqrt{\frac{\tilde{d}}{l_p}}\|\mathrm{Proj}_{V_i}(\boldsymbol{x}_t)\| \geq \left(1 - \frac{1}{2k}\right)\|\mathrm{Proj}_{E_p}(\boldsymbol{x}_t)\|.$$

Hence, if $\boldsymbol{x}_t$ is a depth-p' query for $p' > p$, we must have $\|\mathrm{Proj}_{V_i}(\boldsymbol{x}_t)\| \leq \delta_i^{(p)}$, which in turns gives

$$\|\mathrm{Proj}_{E_p}(\boldsymbol{x}_t)\| \leq \frac{\delta_i^{(p)}}{1 - \frac{1}{2k}}\sqrt{\frac{\tilde{d}}{l_p}} \leq 2\delta_k^{(p)}\sqrt{\frac{\tilde{d}}{l_p}} = \eta_p.$$

Hence, under $\mathcal{G}$, all claims hold and we have

$$\mathbb{P}(\mathcal{G}) \geq 1 - \sum_{i\in[k-1]} \mathbb{P}(\mathcal{F}_i^c \mid \mathcal{E}_i) \geq 1 - k^2 e^{-k\ln d}.$$

We now recall that all the previous discussion was dependent on the choice of $j \in [J_P]$. Taking the union bound over all these choices shows that all claims from the lemma hold with probability at least $1 - k^2 J_P e^{-k\ln d}$. ■

9.5.2 Query lower bounds for an Adapted Orthogonal Subspace Game

By Lemma 9.10, to receive new information about E_p, the algorithm needs to find robustly-independent queries. We next show that we can relate a run of Game 9.13 to playing an instance of some orthogonal subspace game. However, the game needs to be adjusted to take into account the degree of liberty from $\hat{j} \in [J_P]$. This yields following Adapted Orthogonal Subspace Game 9.14.

Input: dimensions d, $\tilde{d}$; memory M; number of robustly-independent vectors k; number of queries m; parameters β, γ; maximum index J

1 *Oracle:* Sample a uniform $\tilde{d}$-dimension linear subspace E in $\mathbb{R}^d$ and i.i.d. vectors $\boldsymbol{v}_r^{(j)} \overset{i.i.d.}{\sim} \mathcal{U}(S_d \cap E)$ for $r \in [m]$ and $j \in [J]$

2 *Player:* Observe E and $\boldsymbol{v}_r^{(j)}$ for all $r \in [m]$ and $j \in [J]$. Based on these, store an M-bit message $\mathsf{Message}$ and an index $\hat{j} \in [J]$

3 *Oracle:* Send samples $\boldsymbol{v}_1^{(\hat{j})}, \dots, \boldsymbol{v}_m^{(\hat{j})}$ to player

4 *Player:* Based on $\mathsf{Message}$ and $\boldsymbol{v}_1^{(\hat{j})}, \dots, \boldsymbol{v}_m^{(\hat{j})}$ only, return unit norm vectors $\boldsymbol{y}_1, \dots, \boldsymbol{y}_k$

5 The player wins if for all $i \in [k]$

1. $\|\mathrm{Proj}_E(\boldsymbol{y}_i)\| \leq \beta$
2. $\|\mathrm{Proj}_{Span(\boldsymbol{y}_1,\dots,\boldsymbol{y}_{i-1})^\perp}(\boldsymbol{y}_i)\| \geq \gamma$.

Game 9.14: Adapted Orthogonal Subspace Game

We first prove a query lower bound for Game 9.14 similar to Theorem 9.4.

Theorem 9.9. *Let $d \geq 8$, $C \geq 1$, and $0 < \beta, \gamma \leq 1$ such that $\gamma/\beta \geq 3e\sqrt{kd/\tilde{d}}$. Suppose that $\frac{\tilde{d}}{4} \geq k \geq 50C\frac{M+2\log_2 J+3}{\tilde{d}} \ln \frac{\sqrt{d}}{\gamma}$. If the player wins at the Adapted Orthogonal Subspace Game 9.14 with probability at least $1/C$, then $m > \frac{\tilde{d}}{2}$.*

Proof Fix parameters satisfying the conditions of the lemma and suppose $m \leq \frac{\tilde{d}}{2}$. In the previous proof for Theorem 9.4, we could directly reduce the Orthogonal Vector Game 9.2 to a simplified version (Game 9.7) in which the query vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ are not present anymore. We briefly recall the construction, which is important for this proof as well. Let E be a uniform $\tilde{d}$-dimensional subspace of $\mathbb{R}^d$ and $\boldsymbol{w}_1, \dots, \boldsymbol{w}_m \overset{i.i.d.}{\sim} \mathcal{U}(S_{d-1} \cap E)$. For any m-dimensional subspace V, let $\mathcal{D}(V)$ be the distribution of $\boldsymbol{w}_1, \dots, \boldsymbol{w}_m$ conditional on $V = Span(\boldsymbol{w}_i, i \in [m])$. We now propose an alternate construction for the subspace E and the vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_m$. Instead, start by sampling a uniform m-dimensional subspace V, then sample $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_m) \sim \mathcal{D}(V)$. Last, let F be a uniform $(\tilde{d}-m)$-dimensional subspace of $\mathbb{R}^d$ independent of all past random variables. We then pose $E := V \oplus F$. We can easily check that the two distributions are equal, and as a result, we assumed without loss of generality that the samples $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ were sampled in that manner. This is convenient because the space F is independent from $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$. Unfortunately, this will not be the case in the

present game for the samples $\boldsymbol{v}_1^{(\hat{j})}, \ldots, \boldsymbol{v}_k^{(\hat{j})}$ because the samples also depend on the index $\hat{j} \in [J]$.

We slightly modify the construction to account for the J different batches of samples.

1. Sample E a uniform $\tilde{d}$-dimensional subspace of $\mathbb{R}^d$
2. Independently for all $j \in [J]$, sample a m-dimensional subspace $V^{(j)}$ of E and sample $(\boldsymbol{v}_1^{(j)}, \ldots, \boldsymbol{v}_k^{(j)}) \sim \mathcal{D}(V^{(j)})$. independently from these random variables, also sample $F^{(j)}$ a uniform $(\tilde{d} - m)$-dimensional subspace of E.

We can check that the construction of the vectors $\boldsymbol{v}_i^{(j)}$ for $i \in [k]$ and $j \in [J]$ is equivalent as that in line 1 of Game 9.14. Further, note that for any fixed $j \in [J]$, the subspaces $V^{(j)}, F^{(j)}$ are independent (because E was also uniformly sampled) and sampled according to uniform m-dimensional (resp. $\tilde{d} - m$-dimensional) subspaces of $\mathbb{R}^d$. Also, on an event $\mathcal{F}_i$ of probability one, $E = V^{(i)} \oplus F^{(i)}$. In particular, their mutual information is zero. We now bound their mutual information for the selected index $\hat{j} \in [J]$ and aim to show that it is at most $\mathcal{O}(\ln J)$. By definition,

$$I(\boldsymbol{V}^{(\hat{j})}; F^{(\hat{j})}) = \int_{\boldsymbol{v}} \int_f p_{\boldsymbol{V}^{(\hat{j})}, F^{(\hat{j})}}(\boldsymbol{v}, f) \ln \frac{p_{\boldsymbol{V}^{(\hat{j})}, F^{(\hat{j})}}(\boldsymbol{v}, f)}{p_{\boldsymbol{V}^{(\hat{j})}}(\boldsymbol{v}) p_{F^{(\hat{j})}}(f)} d\boldsymbol{v} df.$$

Since for any $j \in [J]$, $\boldsymbol{V}^{(j)}$ and $F^{(j)}$ are independent, we have

$$\begin{aligned} p_{\boldsymbol{V}^{(\hat{j})}, F^{(\hat{j})}}(\boldsymbol{v}, f) = \sum_{j \in [J]} p_{\boldsymbol{V}^{(j)}, F^{(j)}, \hat{j}}(\boldsymbol{v}, f, j) &= \sum_{j \in [J]} p_{\boldsymbol{V}, F}(\boldsymbol{v}, f) \mathbb{P}(\hat{j} = j \mid \boldsymbol{V}^{(j)} = \boldsymbol{v}, F^{(j)} = f) \\ &= p_{\boldsymbol{V}}(\boldsymbol{v}) p_F(f) \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid \boldsymbol{V}^{(j)} = \boldsymbol{v}, F^{(j)} = f). \end{aligned}$$

Similarly, we have

$$p_{\boldsymbol{V}^{(\hat{j})}}(\boldsymbol{v}) = p_{\boldsymbol{V}}(\boldsymbol{v}) \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid \boldsymbol{V}^{(j)} = \boldsymbol{v}) \tag{9.40}$$

$$p_{F^{(\hat{j})}}(f) = p_F(f) \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid F^{(j)} = f). \tag{9.41}$$

For convenience we introduce the following notations

$$\begin{aligned} P_{\boldsymbol{V}, F}(\boldsymbol{v}, f) &:= \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid \boldsymbol{V}^{(j)} = \boldsymbol{v}, F^{(j)} = f) \\ P_{\boldsymbol{V}}(\boldsymbol{v}) &:= \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid \boldsymbol{V}^{(j)} = \boldsymbol{v}) \\ P_F(f) &:= \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid F^{(j)} = f). \end{aligned}$$

Putting the previous equations together gives

$$I(\boldsymbol{V}^{(\hat{j})}; F^{(\hat{j})}) = \int_{\boldsymbol{v}} \int_f p_{\boldsymbol{V}^{(\hat{j})}, F^{(\hat{j})}}(\boldsymbol{v}, f) \ln \frac{P_{\boldsymbol{V}, F}(\boldsymbol{v}, f)}{P_{\boldsymbol{V}}(\boldsymbol{v}) P_F(f)} d\boldsymbol{v} df.$$

We decompose the logarithmic term on the right-hand side and bound each corresponding term separately. We start with the term involving $P_{\boldsymbol{V},F}(\boldsymbol{v},f)$. Define the random variable $\tilde{j}(\boldsymbol{v},f)$ on $[J]$ that has $\mathbb{P}(\tilde{j}(\boldsymbol{v},f)=j)=\mathbb{P}(\hat{j}=j \mid \boldsymbol{V}^{(j)}=\boldsymbol{v}, F^{(j)}=f)/P_{\boldsymbol{V},F}(\boldsymbol{v},f)$. Because $H(\tilde{j}(\boldsymbol{v},f)) \le \ln J$ for all choices of $\boldsymbol{v}, f$ we obtain

$$\begin{aligned}
&\int_{\boldsymbol{v}}\int_{f} p_{\boldsymbol{V}^{(\hat{j})},F^{(\hat{j})}}(\boldsymbol{v},f)\ln P_{\boldsymbol{V},F}(\boldsymbol{v},f)d\boldsymbol{v}df \\
&= \int_{\boldsymbol{v}}\int_{f} p_{\boldsymbol{V}}(\boldsymbol{v})p_F(f)\sum_{j\in[J]}\mathbb{P}(\hat{j}=j \mid \boldsymbol{V}^{(j)}=\boldsymbol{v}, F^{(j)}=f)\ln P_{\boldsymbol{V},F}(\boldsymbol{v},f)d\boldsymbol{v}df \\
&\le \int_{\boldsymbol{v}}\int_{f} p_{\boldsymbol{V}}(\boldsymbol{v})p_F(f)\sum_{j\in[J]}\mathbb{P}(\hat{j}=j \mid \boldsymbol{V}^{(j)}=\boldsymbol{v}, F^{(j)}=f)\ln \frac{P_{\boldsymbol{V},F}(\boldsymbol{v},f)}{\mathbb{P}(\hat{j}=j \mid \boldsymbol{V}^{(j)}=\boldsymbol{v}, F^{(j)}=f)}d\boldsymbol{v}df \\
&\le \ln J \int_{\boldsymbol{v}}\int_{f} p_{\boldsymbol{V}}(\boldsymbol{v})p_F(f)\sum_{j\in[J]}\mathbb{P}(\hat{j}=j \mid \boldsymbol{V}^{(j)}=\boldsymbol{v}, F^{(j)}=f)d\boldsymbol{v}df = \ln J.
\end{aligned}$$

We next turn to the term involving $P_{\boldsymbol{V}}(\boldsymbol{v})$. Since $x\ln\frac{1}{x} \le 1/e$ for all $x \ge 0$, we have directly

$$\begin{aligned}
\int_{\boldsymbol{v}}\int_{f} p_{\boldsymbol{V}^{(\hat{j})},F^{(\hat{j})}}(\boldsymbol{v},f)\ln\frac{1}{P_{\boldsymbol{V}}(\boldsymbol{v})}d\boldsymbol{v}df &= \int_{\boldsymbol{v}} p_{\boldsymbol{V}^{(\hat{j})}}(\boldsymbol{v})\ln\frac{1}{P_{\boldsymbol{V}}(\boldsymbol{v})}d\boldsymbol{v} \\
&= \int_{\boldsymbol{v}} p_{\boldsymbol{V}}(\boldsymbol{v})P_{\boldsymbol{V}}(\boldsymbol{v})\ln\frac{1}{P_{\boldsymbol{V}}(\boldsymbol{v})}d\boldsymbol{v} \le \frac{1}{e}.
\end{aligned}$$

We similarly get the same bound for the term involving $P_F(f)$. Putting these estimates together we finally obtain

$$I(\boldsymbol{V}^{(\hat{j})};F^{(\hat{j})}) \le \ln J + \frac{2}{e} \le \ln J + 1. \tag{9.42}$$

From now, we use similar arguments as in the proof of Lemma 9.4. We fix $s = 1 + \ln\frac{\sqrt{d}}{\gamma}$. Denote by $\mathcal{E}$ the event when the player wins and denote by $\boldsymbol{Y} = [\boldsymbol{y}_1,\dots,\boldsymbol{y}_k]$ the concatenation of the vectors output by the player. Using Lemma 9.5 and the same arguments as in Lemma 9.4, we construct from $\boldsymbol{Y}$ an orthonormal sequence $\boldsymbol{Z} = [\boldsymbol{z}_1,\dots,\boldsymbol{z}_r]$ with $r = \lceil k/s \rceil$ such that on the event $\mathcal{E}$, for all $i \in [r]$,

$$\|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \le \frac{e\beta\sqrt{k}}{\gamma} \le \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}}.$$

In particular, we have

$$\|\mathrm{Proj}_{F^{(\hat{j})}}(\boldsymbol{z}_i)\| \le \|\mathrm{Proj}_E(\boldsymbol{z}_i)\| \le \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}}, \quad i \in [r] \tag{9.43}$$

We now give both an upper and lower bound on $I(F^{(\hat{j})};\boldsymbol{Y})$ by adapting the arguments from Lemma 9.4. The data processing inequality gives

$$\begin{aligned}
I(F^{(\hat{j})};\boldsymbol{Y}) \le I(F^{(\hat{j})};\mathsf{Message},\boldsymbol{V}^{(\hat{j})}) &= I(F^{(\hat{j})};\boldsymbol{V}^{(\hat{j})}) + I(F^{(\hat{j})};\mathsf{Message} \mid \boldsymbol{V}^{(\hat{j})}) \\
&\le M\ln 2 + \ln J + 1.
\end{aligned}$$

In the last inequality, we used the fact that Message is encoded in M bits (to avoid continuous/discrete issues with the mutual information, this step can be fully formalized as done in the proof of Lemma 9.4) and Eq (9.42).

We next turn to the lower bound. The same arguments as in Lemma 9.4 give

$$I(F^{(\hat{j})}; \boldsymbol{Y}) \geq \mathbb{P}(\mathcal{E}) \mathbb{E}_{\mathcal{E}} \left[I(F^{(\hat{j})}; \boldsymbol{Z} \mid \mathcal{E}) \right] - \ln 2.$$

Next, denote by $\mathcal{C}$ the set of $(\tilde{d}-m)$-dimensional subspaces F compatible with Eq (9.43), that is

$$\mathcal{C} := \mathcal{C}(\boldsymbol{Z}) = \left\{ (\tilde{d}-m)\text{-dimensional subspace } F \text{ of } \mathbb{R}^d : \|\mathrm{Proj}_F(\boldsymbol{z}_i)\| \leq \frac{1}{3}\sqrt{\frac{\tilde{d}}{d}}, i \in [r] \right\}.$$

The same arguments as in Lemma 9.4 (see Eq (9.28)) show that

$$I(F^{(\hat{j})}; \boldsymbol{Z} \mid \mathcal{E}) \geq \mathbb{E}_{\mathcal{C}|\mathcal{E}} \left[\ln \frac{1}{\mathbb{P}(\hat{F}^{(j)} \in \mathcal{C})} \right] + \ln \mathbb{P}(\mathcal{E}).$$

The rest of the proof of Lemma 9.4 shows that letting G be a random uniform $(\tilde{d}-m)$-dimensional subspace of $\mathbb{R}^d$, for any realization of $\boldsymbol{Z}$, we have

$$\mathbb{P}(G \in \mathcal{C}(\boldsymbol{Z})) \leq e^{-(\tilde{d}-m)r/16}.$$

While $F^{(\hat{j})}$ is not distributed as a uniform $(\tilde{d}-m)$-dimensional subspace of $\mathbb{R}^d$, we show that is close from it. Indeed, for any choice of $(\tilde{d}-m)$-dimensional subspace f, by Eq (9.41) we have

$$p_{F^{(\hat{j})}}(f) = p_F(f) \sum_{j \in [J]} \mathbb{P}(\hat{j} = j \mid F^{(j)} = f) \leq J p_F(f),$$

where $F \sim G$ is distributed a uniform $(\tilde{d}-m)$-dimensional subspace of $\mathbb{R}^d$. Thus, for any realization of $\boldsymbol{Z}$, we obtain

$$\mathbb{P}(F^{(\hat{j})} \in \mathcal{C}(\boldsymbol{Z})) \leq J \cdot \mathbb{P}(F \in \mathcal{C}(\boldsymbol{Z})) = J \cdot \mathbb{P}(G \in \mathcal{C}(\boldsymbol{Z})) \leq J e^{-(\tilde{d}-m)r/16}.$$

Putting together all the previous equations, we obtained

$$\begin{aligned} M \ln 2 + \ln J + 1 \geq I(F^{(\hat{j})}; \boldsymbol{Y}) &\geq \mathbb{P}(\mathcal{E}) \left(\frac{(\tilde{d}-m)r}{16} - \ln J \right) - \mathbb{P}(\mathcal{E}) \ln \frac{1}{\mathbb{P}(\mathcal{E})} - \ln 2 \\ &\geq \mathbb{P}(\mathcal{E}) \frac{\tilde{d}r}{32} - \ln J - \ln 2 - \frac{1}{e}. \end{aligned}$$

As a result, using $d \geq 8$ so that $s \leq 2 \ln \frac{\sqrt{d}}{\gamma}$, we have

$$\mathbb{P}(\mathcal{E}) \leq 32 \frac{M \ln 2 + 2 \ln J + 3 \ln 2}{\tilde{d}r} \leq 50 \frac{M + 2 \log_2 J + 3}{\tilde{d}k} \ln \frac{\sqrt{d}}{\gamma}.$$

In the last inequality, we used the assumption on k. We obtain a contradiction, which shows that $m \geq \frac{\tilde{d}}{2}$. ■

9.5.3 Recursive lower bounds for the feasibility game and feasibility problems

We can now start the recursive argument to give query lower bounds. Precisely, we relate the feasibility Game 9.13 to the Adapted Orthogonal Subspace Game 9.14. The intuition is similar as in the deterministic case except for a main subtlety. We do not restart depth-$(P-1)$ periods once k exploratory queries have been performed. As a result, it may be the case that for such a period after having found k exploratory queries, the algorithm gathers a lot of information on the next layer P without having to perform new exploratory queries. This is taken into account in the following result.

Lemma 9.11. *Let $d \geq 8$ and $\zeta \geq 1$. Suppose that $\frac{\tilde{d}}{4l_{P-1}} \geq k \geq 100\zeta \frac{M+2\log_2 J_P+3}{\tilde{d}} \ln \frac{\sqrt{d}}{\gamma_P}$, that $k \ln d \geq \ln(2\zeta k^2 J_P(N+1))$, and $k \geq 3$. Also, suppose that $l_P, l_{P-1} \geq Ck^3 \ln d$. If that there exists a strategy for Game 9.13 with P layers and maximum index J_P that wins with probability at least q, then there exists a strategy for Game 9.13 with $P-1$ layers that wins with probability at least $q - \frac{1}{2\zeta}$ with same parameters as the depth-P game for $(d, \tilde{d}, l, k, M, T_1, \ldots, T_{P-1})$ and maximum index $J_{P-1} = NJ_P$.*

Proof Fix $P \geq 2$ and a strategy for Game 9.13 with P layers. Within this proof, to simplify the notations, we will write l instead of l_{P-1}. In fact, this is consistent with the parameters that were specified at the beginning of the section ($l_p = l$ for all $p \in [P-1]$). Using this strategy, we construct a strategy for Game 9.14 for the parameters $m = Nlk \leq \frac{\tilde{d}}{2}$ and $J = J_P$ and memory limit $M + \log_2 J + 1$.

The strategy for the orthogonal subspace game is described in Algorithm 9.15. Similarly as the strategy constructed for the deterministic case (Algorithm 9.9), it uses the samples $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{lkN}$ provided by the oracle to construct the depth-$(P-1)$ probing subspaces using l new vectors for each new subspace. The only subtlety is that in line 2 of Game 9.14 the player needs to select the index $\hat{j} \in [J]$ guiding the samples that will be received in line 4 of Game 9.14. The knowledge of $\hat{j}$ is necessary to resample the probing subspaces for depths $p \neq P-1$ (see line 5 of Algorithm 9.15) hence we also add it to the message. The message can therefore be encoded into $M + \lceil \log_2 J \rceil \leq M + \log_2 J + 1$ bits.

We next define some events under which we will show that Algorithm 9.15 wins at Game 9.14. We introduce the event in which the algorithm makes at most k exploratory queries for any of the N depth-$(P-1)$ periods:

$$\mathcal{E} = \bigcap_{a<N} \{\text{at most } k \text{ depth-}(P-1) \text{ exploratory queries during } [aT_{P-1}, (a+1)T_{P-1})\}.$$

Next, by Lemma 9.10 and the union bound, on an event $\mathcal{F}$ of probability at least $1 - k^2 J_P(N+1)e^{-k\ln d}$, the exploration of the probing subspaces for the N depth-$(P-1)$ periods and the only depth-P period, all satisfy the properties listed in Lemma 9.10. In particular, under $\mathcal{E} \cap \mathcal{F}$, all depth-$(P-1)$ periods are proper in the following sense: if $\boldsymbol{x}_t$ for $t \in [T_P]$ is a depth-P query then because there were at most $r_{P-1}(t) \leq k$ exploratory queries in the corresponding depth-$(P-1)$ period, we have

$$\|\mathrm{Proj}_{E_{P-1}}(\boldsymbol{x}_t)\| \leq \eta_{P-1}. \tag{9.44}$$

Input: depth P, dimensions d, $\tilde{d}$, number of vectors k, M-bit algorithm alg for Game 9.13 with P layers; T_P; maximum index J_P

Part 1: Construct the message and index

1 For all $p \in [P] \setminus \{P-1\}$, sample independently E_p a uniform $\tilde{d}$-dimensional linear subspaces in $\mathbb{R}^d$, and J_P i.i.d. sequences $\mathbb{V}^{(p,1)}, \dots, \mathbb{V}^{(p,J_P)}$ as in line 2 of Game 9.13

2 Observe E and vectors $\boldsymbol{v}_r^{(j)}$ for $r \in [Nlk]$ and $j \in [J_P]$. Set $E_{p-1} = E$ and for all $j \in [J_P]$ let $\mathbb{V}^{(P-1,j)} := (\boldsymbol{V}^{(P-1,j,a)})_{a \in [0,N)}$ where $\boldsymbol{V}^{(P-1,j,a)} := (Span(\boldsymbol{v}^{(j)}_{alk+(i-1)l+s}, s \in [l]))_{i \in [k]}$ for $a \in \{0, \dots, N-1\}$. Given all previous information, construct the M-bit message Memory and the index $\hat{j} \in [J_P]$ as in line 3 of Game 9.13

3 Submit to the oracle the message $\mathsf{Message} = (\mathsf{Memory}, \hat{j})$ and the index $\hat{j}$

Part 2: Simulate run of Game 9.13

4 Receive $\mathsf{Message} = (\mathsf{Memory}, \hat{j})$ and samples $\boldsymbol{v}_1^{(\hat{j})}, \dots, \boldsymbol{v}_{Nlk}^{(\hat{j})}$ from Oracle. Based on the samples, construct $\mathbb{V}^{(p,\hat{j})}$ from these samples as in Part 1

5 For $p \in [P] \setminus \{P-1\}$, resample E_p and $\mathbb{V}^{(p,\hat{j})}$ using same randomness as in Part 1 and knowledge of $\hat{j}$.

6 Initialize memory of alg to Memory and run T_P iterations of the feasibility problem using $\mathbb{V}^{(p,\hat{j})}$ for $p \in [P]$ as in lines 5-9 of Game 9.13

7 **if** *there were less than k depth-P exploratory queries* **then** Strategy fails; **end**;

8 **return** normalized depth-P exploratory queries $\frac{\boldsymbol{y}_1^{(P)}}{\|\boldsymbol{y}_1^{(P)}\|}, \dots, \frac{\boldsymbol{y}_k^{(P)}}{\|\boldsymbol{y}_k^{(P)}\|}$

Algorithm 9.15: Strategy of the Player for the Adapted Orthogonal Subspace Game 9.14

Finally, let $\mathcal{G}$ be the event on which the strategy wins. Suppose that $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$ is satisfied. Because the strategy wins, we know that the oracle used the last depth-P probing subspace $V_k^{(P)}$. Because $\mathcal{F}$ is satisfied, in turn, this shows that there were at least k depth-P exploratory queries. In particular, the strategy from Algorithm 9.15 does not fail. We recall that exploratory queries $\boldsymbol{y}_i^{(P)}$ for $i \in [k]$ must satisfy $\|\boldsymbol{y}_i^{(P)}\| \geq \frac{1}{2}$ since $\boldsymbol{e}^\top \boldsymbol{y}_i^{(P)} \leq -\frac{1}{2}$. Then, by Eq (9.44) writing $\boldsymbol{u}_i := \boldsymbol{y}_i^{(P)} / \|\boldsymbol{y}_i^{(p)}\|$ for $i \in [k]$, we obtain

$$\|\mathrm{Proj}_E(\boldsymbol{u}_i)\| = \|\mathrm{Proj}_{E_{P-1}}(\boldsymbol{u}_i)\| \leq 2\|\mathrm{Proj}_{E_{P-1}}(\boldsymbol{y}_i^{(P)})\| \leq 2\eta_{P-1}, \quad i \in [k].$$

Last, by definition, the exploratory queries are robustly-independent:

$$\|\mathrm{Proj}_{Span(\boldsymbol{u}_j^{(p)}, j<i)^\perp}(\boldsymbol{u}_i^{(p)})\| \geq \|\mathrm{Proj}_{Span(\boldsymbol{y}_j^{(p)}, j<i)^\perp}(\boldsymbol{y}_i^{(p)})\| \geq \gamma_P, \quad j \in [k].$$

In summary, on $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, the algorithm wins at Game 9.14 using memory at most $M + \log_2 J + 1$, $m = Nlk \leq \frac{\tilde{d}}{2}$ queries and parameters $(\beta, \gamma) = (2\eta_{P-1}, \gamma_P)$. It suffices to check that the assumptions from Theorem 9.9 are satisfied. We only need to check the bound on γ/β. We compute

$$\frac{\gamma}{\beta} = \frac{\gamma_P}{2\eta_{P-1}} = \frac{\delta_1^{(p)}}{8k\eta_{P-1}} \geq \frac{\mu}{16k}\left(1 - \frac{2}{k}\right)^{k-1}\sqrt{\frac{l}{\tilde{d}}} \geq 3e\sqrt{\frac{kd}{\tilde{d}}}.$$

In the last inequality, we used the fact that because $k \geq 3$, we have $(1-2/k)^{k-1} \geq (1-2/3)^2$. Combining the previous results, we obtain

$$\mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \leq \mathbb{P}(\text{Algorithm 9.15 wins at Game 9.14}) \leq \frac{1}{2\zeta}.$$

In turn, this shows that

$$\begin{aligned}\mathbb{P}(\mathcal{E}^c) &\geq \mathbb{P}(\mathcal{E}^c \cap \mathcal{F} \cap \mathcal{G}) = \mathbb{P}(\mathcal{F} \cap \mathcal{G}) - \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{F}) \\ &\geq \mathbb{P}(\mathcal{G}) - \mathbb{P}(\mathcal{F}^c) - \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \\ &\geq q - k^2 J_P (N+1) e^{-k \ln d} - \frac{1}{2\zeta} \geq q - \frac{1}{2\zeta} - \frac{1}{2\zeta} = q - \frac{1}{\zeta}.\end{aligned}$$

The last step of the proof is to construct a strategy for Game 9.13 with $P-1$ layers that wins under the event $\mathcal{E}^c$. Note that this event corresponds exactly to the case when in at least one of the depth-$(P-1)$ periods the algorithm performed k exploratory queries. Hence, the strategy for $P-1$ layers mainly amounts to simulating that period. This can be done thanks to the index $\hat{j}$ which precisely specifies the probing subspaces needed to simulate that winning period. Because there were N depth-$(P-1)$ periods, the new index parameter becomes $J_{P-1} := NJ_P$. The complete strategy is described in Algorithm 9.16 and is similar to the one constructed for the deterministic case in Lemma 9.7 (Algorithm 9.10). The main difference with the deterministic case is that there is no need to keep exploratory queries for larger depths (here there is only P) in the new strategy. Indeed, because the oracle is non-adaptive, these deeper layers do not provide information on the other layers.

It is straightforward to check that the sequences $\mathbb{W}^{(p,1)}, \ldots, \mathbb{W}^{(p,J_P)}$ for $p \in [P]$ constructed in Algorithm 9.16 are identically distributed as the sequences constructed by the oracle of Game 9.13 for P layers. By definition of $\mathcal{E}$, under $\mathcal{E}^c$ there is a depth-$(P-1)$ period with k exploratory queries hence the strategy does not fail. Because of the choice of index $(\hat{j}-1)N + \hat{a} + 1$, during the run lines 5-9 of Game 9.13 for depth $P-1$, (since *alg* reuses exactly the same randomness) the oracle exactly implements the $\hat{a}$-th depth-$(P-1)$ period. Hence under $\mathcal{E}^c$, Algorithm 9.16 wins. This ends the proof. ■

Applying the previous result for Game 9.13 with all depths $p \in \{2, \ldots, P\}$ recursively gives the following query lower bound.

Theorem 9.10. *Let $P \geq 2$ and $d \geq 40P$. Suppose that $l_P \geq l$ and*

$$c_5 \frac{M + P \ln d}{d} P^3 \ln d \leq k \leq c_4 \left(\frac{d}{P \ln d} \right)^{1/4} \tag{9.45}$$

for some universal constants $c_4, c_5 > 0$. If a strategy for Game 9.4 for depth P and maximum index $J_P = N$ uses M bits of memory and wins with probability at least $\frac{1}{2}$, then it performed at least

$$T_{max} > T_P \geq \frac{k}{2} \left(\frac{d}{12Plk} \right)^{P-1}$$

queries.

Input: dimensions d, $\tilde{d}$, number of vectors k, depth p, M-bit memory algorithm alg for Game 9.13 for P layers; maximum index J_P

Output: strategy for Game 9.13 for $P-1$ layers with maximum index $J_{P-1} = NJ_P$

1 Receive subspaces $E_1, \dots, E_{P-1}$ and sequences $\mathbb{V}^{(p,1)}, \dots, \mathbb{V}^{(p,J_{P-1})}$ for $p \in [P-1]$

2 Sample E_P as an independent uniform $\tilde{d}$-dimensional subspace of $\mathbb{R}^d$ and sample i.i.d. sequences $\mathbb{W}^{(P,1)}, \dots, \mathbb{W}^{(P,J_P)}$ as in line 2 of Game 9.13 for P layers

3 For $p \in [P-1]$, reorganize the sequences $\mathbb{V}^{(p,j)}$ as follows. For $j \in [J_P]$ let $\mathbb{W}^{(p,j)}$ be the concatenation of $\mathbb{V}^{(p,N(j-1)+1)}, \dots, \mathbb{V}^{(p,Nj)}$ in that order so that $\mathbb{W}^{(p,j)}$ has N^{P-p} elements

4 Based on all E_p and $\mathbb{W}^{(p,j)}$ for $j \in [J_P]$ and $p \in [P]$ initialize memory of alg to the M-bit message Memory and store the index $\hat{j} \in [J_P]$ as in line 3 of Game 9.13 with P layers

5 Run T_P iterations of feasibility problem with alg using $\mathbb{W}^{(1,\hat{j})}, \dots, \mathbb{W}^{(P,\hat{j})}$ as in lines 5-9 of Game 9.13.

6 **if** *at most* $k-1$ *exploratory queries for all depth-*$(P-1)$ *periods* **then** Strategy fails; **end**;

7 **else**

8 Let $[\hat{a}T_{P-1}, (\hat{a}+1)T_{P-1})$ be the first such depth-$(P-1)$ period for $\hat{a} \in [0, N)$

9 Submit memory state Message of alg at the beginning of the period (just before iteration $\hat{a}T_{P-1}$) and index $(\hat{j}-1)N + \hat{a} + 1 \in [J_{P-1}]$

10 **end**

Algorithm 9.16: Strategy of the Player for Game 9.13 with $P-1$ layers given a strategy for P layers

Proof Suppose for now that the parameter k satisfies all assumptions from Lemma 9.11 for all Games 9.13 with layers $p \in \{2, \dots, P\}$ and $\zeta = P$. Then, starting from a strategy for depth P, maximum index $J_P = N$, and winning with probability q with T_P queries, we can construct a strategy for the depth-1 game with maximum index $J_1 = N^P$ and wins with probability at least $q - \frac{1}{2P}(P-1) \geq q - \frac{P-1}{2P} > 0$. As in the deterministic case, to win at Game 9.13 for depth 1 one needs at least k queries and we reach a contradiction since $T_1 = \frac{k}{2}$. Hence, this shows that an algorithm that wins with probability at least $\frac{1}{2}$ at Game 9.13 with P layers and $J_P = N$ must make at least the following number of queries

$$T_{max} > T_P = \frac{k}{2} N^{P-1}.$$

Now assuming that $\tilde{d} \geq 4lk$, we obtain $N \geq \tilde{d}/(4lk)$. Hence, using $\tilde{d} \geq \frac{d}{3P}$, we obtain the desired lower bound

$$T_{max} > \frac{k}{2} \left\lfloor \frac{d}{12Plk} \right\rfloor^{P-1}.$$

We now check that the assumptions for Lemma 9.11 are satisfied for all games with layers $p \in \{2, \dots, P\}$. It suffices to check that

$$\frac{\tilde{d}}{4l} \geq k \geq 100P \frac{M + 2\log_2 J_1 + 3}{\tilde{d}} \ln \frac{\sqrt{d}}{\gamma_1} \vee \frac{\ln(2Pk^2 J_1(N+1))}{\ln d} \vee 3. \tag{9.46}$$

For the upper bound, recalling the definition of l in Eq (9.33), we have that

$$\frac{\tilde{d}}{4l} \geq \frac{d}{12Pl} = \Omega\left(\frac{d}{k^3 P \ln d}\right).$$

In particular, the left-hand-side of Eq (9.46) holds for

$$k \leq \Omega\left(\left(\frac{d}{P \ln d}\right)^{1/4}\right).$$

For the upper bound, because $\log_2 J_1 \leq P \log_2 d$ and $\gamma_1 = \frac{\delta_1^{(1)}}{4k}$, we have

$$100P\frac{M + 2\log_2 J_1 + 3}{\tilde{d}} \ln \frac{\sqrt{d}}{\gamma_1} = \mathcal{O}\left(\frac{M + P \ln d}{d} P^3 \ln d\right).$$

On the other hand, $\ln(2Pk^2 J_P(N+1)) = \mathcal{O}(P \ln d)$, hence the first term in the right-hand side of Eq (9.46) dominates. As a summary, for an appropriate choice of constants $c_4, c_5 > 0$, Eq (9.46) holds, which ends the proof. ■

The last step of the proof is to link the feasibility problem with the oracle $\tilde{\mathcal{O}}_t$ for $t \geq 0$ with Game 9.13. This step is exactly similar to the deterministic case when we reduced Procedure 9.3 to Game 9.4. By giving a reduction to the Kernel Discovery Game 9.11 we show that an algorithm that solves the feasibility problem with the oracles $\tilde{\mathcal{O}}_t$ must solve multiple instances of Game 9.13 with P layers.

Lemma 9.12. *Let $P \geq 2$ and $k \geq 3$. Suppose that $4l_P k \leq \tilde{d}$ and $l_P \geq l$. Let $N_P = \left\lfloor \tilde{d}/(2l_P k) \right\rfloor$. Suppose that there is an M-bit algorithm that solves the feasibility problem with the randomized oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ using at most M bits of memory and $N_P T_P$ iterations, and that finds a feasible solution with probability at least q. Then, there exists a strategy Game 9.13 for depth P and maximum index $J_P = N$ that uses M bits of memory, T_P iterations and wins with probability at least $q - k^2 N_P e^{-k \ln d} - e^{-\tilde{d}/10}$.*

Proof Fix an M-bit feasibility algorithm alg for the randomized oracle $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ satisfying the hypothesis. We construct from this algorithm a strategy for the Kernel Discovery Game 9.11 with $m = N l_P k \leq \frac{\tilde{d}}{2}$ samples. The construction is essentially the same as that for the deterministic case in Lemma 9.9 (Algorithm 9.12): we simulate a run of the feasibility problem using for E_P the $\tilde{d}$-dimensional space sampled by the oracle and using the samples of the oracle $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ to construct the depth-P probing subspaces. The strategy is given in Algorithm 9.17.

The rest of the proof uses the same arguments as for Lemma 9.11. Define the event

$$\mathcal{E} = \bigcap_{a<N} \{\text{at most } k \text{ depth-}P \text{ exploratory queries during period } [aT_P, (a+1)T_P)\}.$$

By Lemma 9.10 and the union bound, on an event $\mathcal{F}$ of probability at least $1 - k^2 N_P e^{-k \ln d}$, all depth-P periods $[aT_P, (a+1)T_P)$ for $a \in [0, N_P)$ satisfy the properties from Lemma 9.10.

Input: depth P, dimensions d, $\tilde{d}$, l, parameter k, M-bit algorithm alg, number of samples $m = Nl_Pk$

1 Sample $E_1, \dots, E_{P-1}$ i.i.d. uniform $\tilde{d}$-dimensional subspaces of $\mathbb{R}^d$ and independent sequences $(\boldsymbol{V}^{(p,a)})_{a\geq 0}$ for $p \in [P-1]$ as in the construction of the randomized oracles $\tilde{\mathcal{O}}_t$.

2 Receive samples $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{Nl_Pk}$. Let $V_i^{(P,a)} := Span(\boldsymbol{v}_{al_Pk+(i-1)l_P+s}, s \in [l_P])$ for $i \in [k], a \in [0, N)$

3 Set memory of alg to $\mathbf{0}$ and run N_PT_P iterations of the feasibility problem with alg and the oracles $\tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}$ for $t \in [0, N_PT_P)$

4 **if** *at any time* $t \in [0, N_PT_P)$ *the oracle outputs* Success *on query* $\boldsymbol{x}_t$ *of* alg **then**

5 | **return** $\frac{\boldsymbol{x}_t}{\|\boldsymbol{x}_t\|}$ to oracle; **break**

6 **else** Strategy fails; **end** ;

Algorithm 9.17: Strategy of the Player for the Kernel Discovery Game 9.11

Last, let $\mathcal{G}$ be the event on which alg solves the feasibility problem with oracles $(\tilde{\mathcal{O}}_t)$ constructed using the same sequences $(\boldsymbol{V}^{(p,a)})_{a\geq 0}$ for $p \in [P]$ as constructed in Algorithm 9.17. Under $\mathcal{E} \cap \mathcal{F}$, any query $\boldsymbol{x}_t$ for $t \in [0, N_PT_P)$ that passed probes at depth P satisfies

$$\|\mathrm{Proj}_{E_P}(\boldsymbol{x}_t)\| \leq \eta_P.$$

Now note that under $\mathcal{G}$, the algorithm run with the oracles $\tilde{\mathcal{O}}_t$ for $t \in [0, N_PT_P)$ finds a successful query for the oracles and in particular there is some query $t \in [0, N_PT_P)$ which passed all probes at all depths $p \in [P]$:

$$\tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}(\boldsymbol{x}_t) = \mathsf{Success}.$$

Consider the first time $\hat{t}$ when a query $\boldsymbol{x}_t$ is successful for the oracles $\tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}$. We pose $\hat{t} = N_PT_P$ if there is no such query. By construction, at all previous times the responses satisfy

$$\tilde{\mathcal{O}}_t(\boldsymbol{x}_t) = \tilde{\mathcal{O}}_{\boldsymbol{V}^{(1,\lfloor t/T_1 \rfloor)},\dots,\boldsymbol{V}^{(P,\lfloor t/T_P \rfloor)}}(\boldsymbol{x}_t), \quad t < \hat{t}.$$

In summary, up until this time $\hat{t}$, the run in line 3 of Algorithm 9.17 is equivalent to a run of the feasibility with the original oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$. Hence, under $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$, the algorithm returns the vector $\boldsymbol{y} := \boldsymbol{x}_{\hat{t}}/\|\boldsymbol{x}_{\hat{t}}\|$ and we have $\|\mathrm{Proj}_{E_P}(\boldsymbol{x}_{\hat{t}})\| \leq \eta_P$. The successful query also satisfies $\|\boldsymbol{x}_{\hat{t}}\| \geq \frac{1}{2}$ since $\boldsymbol{e}^\top \boldsymbol{x}_{\hat{t}} \leq -\frac{1}{2}$. As a result, under $\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}$,

$$\|\mathrm{Proj}_E(\boldsymbol{y})\| = \|\mathrm{Proj}_{E_P}(\boldsymbol{y})\| \leq 2\|\mathrm{Proj}_{E_P}(\boldsymbol{x}_{\hat{t}})\| \leq 2\eta_P \leq \sqrt{\frac{\tilde{d}}{20d}}.$$

Because $m = Nl_Pk \leq \frac{\tilde{d}}{2}$, Lemma 9.8 implies

$$\mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) \leq \mathbb{P}(\text{Algorithm 9.17 wins at Game 9.11}) \leq e^{-\tilde{d}/10}.$$

Hence,

$$\mathbb{P}(\mathcal{E}^c) \geq \mathbb{P}(\mathcal{G}) - \mathbb{P}(\mathcal{E} \cap \mathcal{F} \cap \mathcal{G}) - \mathbb{P}(\mathcal{F}^c) \geq q - k^2 N_P e^{-k \ln d} - e^{-\tilde{d}/10}.$$

From this, using the exact same arguments as in Lemma 9.11 (Algorithm 9.16) we can construct a strategy for Game 9.13 with P layers and maximal index $J_P = N_P$, and wins under the event $\mathcal{E}^c$. It suffices to simulate the specific depth-P period on which alg queried the successful vector $\boldsymbol{x}_{\hat{t}}$. Note that because $l_P \geq l$, we have $N_P \leq N$. Hence, this strategy also works if instead we have access to a larger maximal index $J_P = N$. This ends the proof. ■

We now combine this reduction to Game 9.13 together with the query lower bound of Theorem 9.10. This directly gives the following result.

Theorem 9.11. *Let $P \geq 2$ and $d \geq 20P$. Suppose that k satisfies Eq (9.45) as in Theorem 9.10. Also, suppose that $4l_P k \leq \tilde{d}$ and $l_P \geq l$. If an algorithm solves the feasibility problem with oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ using M bits of memory and at most T_{max} queries with probability at least $\frac{3}{4}$, then*

$$T_{max} > N_P T_P \geq \frac{kl}{2l_P} \left(\frac{d}{12Plk} \right)^P.$$

Proof Fix parameters satisfying Eq (9.45). Suppose that we have such an algorithm alg for the feasibility problem with oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ that only uses $T_{max} \leq N_P T_P$ queries and wins with probability at least $\frac{3}{4}$. Because $k^2 N_P e^{-k \ln d} + e^{-\tilde{d}/10} \leq \frac{1}{4}$, we can directly combine Lemma 9.12 with Theorem 9.10 to reach a contradiction. Hence, $T_{max} > N_P T_P$ which ends the proof. ■

Now, we easily check that the same proof as Lemma 9.1 shows that under the choice of parameters, with probability at least $1 - e^{-d/40}$ over the randomness of $E_1, \ldots, E_P$, the feasible set $\tilde{Q}_{E_1,\ldots,E_P}$ contains a ball of radius $\epsilon = \delta_1^{(1)}/2$, hence using the oracles $(\tilde{\mathcal{O}}_t)_{t\geq 0}$ is consistent with a feasible problem with accuracy ϵ. These observations give the following final query lower bound for memory-constrained feasibility algorithms.

Theorem 9.12. *Let $d \geq 1$ and an accuracy $\epsilon \in (0, \frac{1}{\sqrt{d}}]$ such that*

$$d^{1/4} \ln^2 d \geq c_6 \frac{M}{d} \ln^{3+1/4} \frac{1}{\epsilon}.$$

for some universal constant $c_6 > 0$. Then, any M-bit randomized algorithm that solves any feasibility problems for accuracy ϵ with probability at least $\frac{9}{10}$ makes at least

$$\left(\frac{d}{M} \right)^2 \frac{1}{\epsilon^{2\phi(d,M,\epsilon)}}$$

queries, where $\phi(d, M, \epsilon) = 1 - 4\frac{\ln \frac{M}{d}}{\ln d} - \mathcal{O}\left(\frac{\ln \frac{\ln(1/\epsilon)}{\ln d}}{\ln d} + \frac{\ln \ln d}{\ln d} \right)$.

Proof We first define

$$\tilde{P} := \frac{2 \ln \frac{1}{\epsilon} - \ln \left(\frac{\ln(1/\epsilon)}{\ln d} + 1 \right) - 15}{\ln d + 15}.$$

We then use the following parameters,

$$P := \left\lfloor \tilde{P} \right\rfloor, \quad k := \left\lceil 3c_4 \frac{M}{d} P^3 \ln d \right\rceil, \quad \text{and} \quad l_P := l \vee \left\lfloor \left(\frac{\tilde{d}}{4k} \right)^{P-\tilde{P}} \right\rfloor.$$

Note that in particular, $P \le P_{max} = \frac{\ln \frac{1}{\epsilon}}{\ln d} + 1$. Taking c_6 sufficiently large, the hypothesis constraint implies in particular $d \ge 20P$.

We assume $P \ge 2$ from now. The same proof as Lemma 9.1 shows that on an event $\mathcal{E}$ of probability $1 - e^{-d/40}$ the feasible set $\tilde{Q}_{E_1,\dots,E_P}$ contains a ball of radius $\delta_1^{(1)}/2$. Note that

$$\delta_1^{(1)} = \frac{(1-2/k)^{k-1}}{20\mu_P \mu^{P-2}} \sqrt{\frac{l}{d}} \ge \frac{6k^{3/2}}{\mu^P} \sqrt{\frac{l_P}{l}} \ge \frac{6k}{\mu^{\tilde{P}}} \sqrt{k \left(\frac{\tilde{d}}{4k\mu^2} \right)^{P-\tilde{P}}},$$

where we used $(1-2/k)^{k-1} \ge (1-2/3)^2$ because $k \ge 3$. Furthering the bounds and using $l \ge Ck^3 \ln d$ from Eq (9.33) gives

$$\frac{\delta_1^{(1)}}{2} \ge \frac{3k}{2\mu^{\tilde{P}}} \sqrt{\frac{\tilde{d}}{\mu^2}} \ge \frac{1}{1400\mu^{\tilde{P}}} \sqrt{\frac{l}{Pk}} \ge \frac{1}{1400\mu^{\tilde{P}}\sqrt{P}} \ge \epsilon.$$

Next, because $l \ge Ck^3 \ln d$, we have $\mu \le 1200\sqrt{d/\ln d}$. Now $\tilde{P}$ was precisely chosen so that

$$\tilde{P} \le \frac{2\ln \frac{1}{\epsilon} - \ln(P_{max}) - 2\ln(1400)}{\ln d - \ln\ln d + 2\ln(1200)}.$$

The previous equations show that under $\mathcal{E}$, the oracles $(\tilde{\mathcal{O}}_t)_{t\ge 0}$ form a valid feasibility problem for accuracy $\epsilon > 0$. In particular, an algorithm solving feasibility problems with accuracy ϵ with T_{max} oracle calls and probability at least $\frac{9}{10}$ would solve the feasibility problem with oracles $(\tilde{\mathcal{O}}_t)_{t\ge 0}$ with probability at least $\frac{9}{10} - \mathbb{P}(\mathcal{E}) \ge \frac{9}{10} - e^{-d/40}$. Because $d \ge 40P \ge 80$, the algorithm wins with probability at least $\frac{3}{4}$. We now check that the assumptions to apply Theorem 9.11 hold. By construction of l_P, we have directly $l_P \ge l$. Further, if Eq (9.45)) is satisfied, we would have in particular $4lk \le \tilde{d}$, hence we would also have $4l_P k \le \tilde{d}$. It now suffices to check that Eq (9.45) is satisfied.

As in the proof of Theorem 9.8, without loss of generality we can assume that $M \ge 2d \ln 1/\epsilon$ since this is necessary to solve even convex optimization problems. Hence, k directly satisfies the right-hand side of Eq (9.45). Next, the assumption gives

$$d^{1/4} \ln^2 d \ge \frac{60c_5}{c_4} \frac{M}{d} \ln^{3+1/4} \frac{1}{\epsilon}.$$

Hence, $c_4 \left(\frac{d}{P \ln d}\right)^{1/4} \ge c_4 \left(\frac{d}{P_{max} \ln d}\right)^{1/4} \ge 6c_5 (M/d) P_{max}^3 \ln d \ge k$. As a result, the right-hand side of Eq (9.45) holds. Then, Theorem 9.11 shows that

$$T_{max} \ge \frac{kl}{2l_P} \left(\frac{d}{12Plk} \right)^P \ge \frac{k}{2(2l)^{P-\tilde{P}}} \left(\frac{d}{12Plk} \right)^{\tilde{P}} \ge \frac{1}{8Ck^2 \ln d} \left(\frac{d}{12Plk} \right)^{\tilde{P}} =: \frac{1}{e^{2\tilde{\phi}(d,M,\epsilon)}}.$$

The above equation also holds even if $P < 2$ (that is, $\tilde{P} \leq 1$) because d iterations are necessary even to solve convex optimization problems. We now simplify

$$\begin{aligned}\tilde{\phi}(d, M, \epsilon) &= -\frac{\ln \frac{M}{d}}{\ln \frac{1}{\epsilon}} + \mathcal{O}\left(\frac{\ln \ln \frac{1}{\epsilon}}{\ln \frac{1}{\epsilon}}\right) + \frac{\tilde{P} \ln \frac{d}{12Plk}}{2 \ln \frac{1}{\epsilon}} \\ &= -\frac{\ln \frac{M}{d}}{\ln \frac{1}{\epsilon}} + \frac{\ln d - 4 \ln \frac{M}{d} - 13 \ln \frac{\ln(1/\epsilon)}{\ln d}}{\ln d} + \mathcal{O}\left(\frac{\ln \ln d}{\ln d}\right).\end{aligned}$$

This ends the proof. ■

In the subexponential regime $\ln \frac{1}{\epsilon} \leq d^{o(1)}$, Theorem 9.12 simplifies to Theorem 9.2.

9.6 Appendix

9.6.1 Concentration inequalities

We first state a standard result on the concentration of normal Gaussian random variables.

Lemma 9.13 (Lemma 1 [LM00]). *Let $n \geq 1$ and define $\boldsymbol{y} \sim \mathcal{N}(0, Id_n)$. Then for any $t \geq 0$,*

$$\begin{aligned}\mathbb{P}(\|\boldsymbol{y}\|^2 \geq n + 2\sqrt{nt} + 2t) &\leq e^{-t}, \\ \mathbb{P}(\|\boldsymbol{y}\|^2 \leq n - 2\sqrt{nt}) &\leq e^{-t}.\end{aligned}$$

In particular, plugging $t = n/2$ and $t = (3/8)^2 n \geq n/8$ gives

$$\begin{aligned}\mathbb{P}(\|\boldsymbol{y}\| \geq 2\sqrt{n}) &\leq e^{-n/2}, \\ \mathbb{P}\left(\|\boldsymbol{y}\| \leq \frac{\sqrt{n}}{2}\right) &\leq e^{-n/8}.\end{aligned}$$

For our work, we need the following concentration for quadratic forms.

Theorem 9.13 ([HW71; RV13]). *Let $\boldsymbol{x} = (X_1, \ldots, x_d) \in \mathbb{R}^d$ be a random vector with i.i.d. components X_i which satisfy $\mathbb{E}[X_i] = 0$ and $\|X_i\|_{\psi_2} \leq K$ and let $\boldsymbol{M} \in \mathbb{R}^{d \times d}$. Then, for some universal constant $c_{hw} > 0$ and every $t \geq 0$,*

$$\begin{aligned}\max\left\{\mathbb{P}\left(\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x} - \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}] > t\right), \mathbb{P}\left(\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}] - \boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x} > t\right)\right\} \\ \leq \exp\left(-c_{hw} \min\left\{\frac{t^2}{K^4 \|\boldsymbol{M}\|_F^2}, \frac{t}{K^2 \|\boldsymbol{M}\|}\right\}\right).\end{aligned}$$

We will only need a restricted version of this concentration bound, for which we can explicit the constant c_{hw}.

Lemma 9.14. *Let P be a projection matrix in $\mathbb{R}^d$ of rank r and let $\boldsymbol{x} \in \mathbb{R}^d$ be a random vector sampled uniformly on the unit sphere $\boldsymbol{x} \sim \mathcal{U}(S_{d-1})$. Then, for any $t > 0$,*

$$\begin{aligned}\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \geq \frac{r}{d}(1+t)\right) &\leq e^{-\frac{r}{8}\min(t, t^2)}, \\ \mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \leq \frac{r}{d}(1-t)\right) &\leq e^{-\frac{r}{4}t^2}.\end{aligned}$$

Also, for $t \geq 1$, we have

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \leq \frac{r}{dt}\right) \leq e^{-\frac{r}{2}\ln(t)+\frac{d}{2e}}.$$

Proof We start with the first inequality to prove. Using the exact same arguments as in Proposition 7.10 from Chapter 7 show that for $t \geq 0$,

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \geq t\right) \leq \exp\left(-\frac{d}{2}D\left(\frac{r}{d} \parallel \frac{r}{d}+t\right)\right).$$

Applying this bound to the projection $I_d - P$ implies the other inequality

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \leq -t\right) \leq \exp\left(-\frac{d}{2}D\left(\frac{d-r}{d} \parallel \frac{d-r}{d}+t\right)\right) = \exp\left(-\frac{d}{2}D\left(\frac{r}{d} \parallel \frac{r}{d}-t\right)\right).$$

It only remains to bound the KL divergence. Consider the function $f(x) = x \in [-\frac{r}{d}, 1-\frac{r}{d}] \mapsto D(\frac{r}{d} \parallel \frac{r}{d}+x)$. Then, $f'(0) = 0$ and

$$f''(x) = \frac{r/d}{(r/d+x)^2} + \frac{1-r/d}{(1-r/d-x)^2} \geq \frac{r/d}{(r/d+x)^2}$$

In particular, if $x \leq 0$, $f''(x) \geq \frac{d}{r}$ so that Taylor's expansion theorem directly gives $f(x) \geq \frac{dx^2}{2r}$. Next, for $|x| \leq \frac{r}{d}$, we have $f''(x) \geq \frac{d}{2r}$. Hence $f(x) \geq \frac{dx^2}{4r}$. Otherwise, if $x \geq \frac{r}{d}$, we have

$$D\left(\frac{r}{d} \parallel \frac{r}{d}+x\right) \geq \int_0^x \frac{r/d(x-u)}{2(r/d+u)^2}du = \frac{x}{2} - \frac{r}{2d}\ln\frac{r/d+x}{r/d} \geq \frac{2-\ln 2}{4}x \geq \frac{x}{4}.$$

In the last inequality, we used $\ln(1+t) \leq \frac{\ln 2}{2}t$ for $t \geq 2$. In summary, we obtained for $t \in [0,1]$,

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \leq -\frac{r}{d}t\right) \leq e^{-\frac{rt^2}{4}}.$$

And for any $t \geq 0$,

$$\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 - \frac{r}{d} \geq \frac{r}{d}t\right) \leq e^{-\frac{r}{8}\min(t,t^2)}.$$

We last prove the third claim of the lemma using the same equation:

$$\begin{aligned}\mathbb{P}\left(\|P(\boldsymbol{x})\|^2 \leq \frac{r}{dt}\right) \leq \exp\left(-\frac{d}{2}D\left(\frac{r}{d} \parallel \frac{r}{dt}\right)\right) &\leq \exp\left(-\frac{d}{2}\left(\frac{r}{d}\ln(t) + (1-p)\ln(1-p)\right)\right) \\ &\leq e^{-\frac{r}{2}\ln(t)+\frac{d}{2e}}.\end{aligned}$$

This ends the proof. ■

The previous result gives concentration bounds for the projection of a single random vector onto a fixed subspace. We now use this result to have concentration bounds on the projection of points from a random subspace sampled uniformly, onto a fixed subspace. Similar bounds are certainly known, in fact, the following lemma can be viewed as a variant of the Johnson–Lindenstrauss lemma. We include the proof for the sake of completeness.

Lemma 9.15. *Let P be a projection matrix in $\mathbb{R}^d$ of rank r and let E be a random s-dimensional subspace of $\mathbb{R}^d$ sampled uniformly. Then, for any $t \in (0,1]$,*

$$\mathbb{P}\left((1-t)\|\boldsymbol{x}\| \leq \sqrt{\frac{d}{r}}\|P(\boldsymbol{x})\| \leq (1+t)\|\boldsymbol{x}\|,\ \forall \boldsymbol{x} \in E\right) \geq 1 - \exp\left(s \ln \frac{Cd}{rt} - \frac{rt^2}{32}\right),$$

for some universal constant $c > 0$.

Proof We use an ϵ-net argument. Let $\epsilon = \frac{t}{2}\sqrt{\frac{r}{d}}$ and construct an ϵ-net Σ of the unit sphere of E, $(E \cap S_{d-1})$ such that each element $\boldsymbol{x} \in \Sigma$ is still distributed as a uniform random unit vector. For instance, consider any parametrization, then rotate the whole space Q again by some uniform rotation. Note that $|\Sigma| \leq (c/\epsilon)^s$ for some universal constant $c > 0$ (e.g. see [Tao23, Lemma 2.3.4]). Combining Lemma 9.14, the union bound, and the observation that $\sqrt{1-t} \geq 1 - t$ and $\sqrt{1+t} \leq 1 + t$, we obtain

$$\mathbb{P}\left(1 - \frac{t}{2} \leq \sqrt{\frac{d}{r}}\|P(\boldsymbol{x})\| \leq 1 + \frac{t}{2},\ \forall \boldsymbol{x} \in \Sigma\right) \geq 1 - |\Sigma| e^{-rt^2/32} \geq 1 - \exp\left(s \ln \frac{cd}{rt} - \frac{rt^2}{32}\right).$$

Denote by $\mathcal{E}$ this event. For unit vector $\boldsymbol{y} \in E \cap S_{d-1}$ there exists $\boldsymbol{x} \in \Sigma$ with $\|\boldsymbol{x} - \boldsymbol{y}\| \leq \epsilon$. As a result, we also have $\|P(\boldsymbol{y}) - P(\boldsymbol{x})\| \leq \epsilon$. Under $\mathcal{E}$, the triangular inequality then shows that

$$1 - t \leq \sqrt{\frac{d}{r}}\|P(\boldsymbol{x})\| \leq 1 + t, \quad \boldsymbol{x} \in E \cap S_{d-1}.$$

For an arbitrary vector $\boldsymbol{x} \in E \setminus \{0\}$, we can then apply the above inequality to $\boldsymbol{x}/\|\boldsymbol{x}\|$, which gives the desired result under $\mathcal{E}$. This ends the proof. ■

Last, we need the following result which lower bounds the smallest singular value for rectangular random matrices.

Theorem 9.14 (Theorem 2.13 [DS01])**.** *Given $m, n \in \mathbb{N}$ with $m \leq n$. Let $\beta = m/n$ and let $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ be a matrix with independent Gaussian $\mathcal{N}(0, 1/n)$ coordinates. The singular values $s_1(\boldsymbol{M}) \leq \ldots \leq s_m(\boldsymbol{M})$ satisfy*

$$\max\left\{\mathbb{P}(s_1(\boldsymbol{M}) \leq 1 - \sqrt{\beta} - t), \mathbb{P}(s_m(\boldsymbol{M}) \geq 1 + \sqrt{\beta} + t)\right\} \leq e^{-nt^2/2}.$$

9.6.2 Decomposition of robustly-independent vectors

Here we give the proof of Lemma 9.5, which is essentially the same as in [Mar+22, Lemma 34] or Lemma 7.6 from Chapter 7.

Proof of Lemma 9.5 Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r)$ be the orthonormal basis given by the Gram-Schmidt decomposition of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r$. By definition of the Gram-Schmidt decomposition, we can write $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{C}$ where $\boldsymbol{C}$ is an upper-triangular matrix. Further, its diagonal is exactly $diag(\|P_{Span(\boldsymbol{y}_{l'}, l'<l)^\perp}(\boldsymbol{y}_l)\|, l \leq r)$. Hence,

$$\det(\boldsymbol{Y}) = \det(\boldsymbol{C}) = \prod_{l \leq r} \|P_{Span(\boldsymbol{y}_{l'}, l'<l)^\perp}(\boldsymbol{y}_l)\| \geq \delta^r.$$

We now introduce the singular value decomposition $\boldsymbol{Y} = \boldsymbol{U} diag(\sigma_1, \ldots, \sigma_r) \boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times r}$ have orthonormal columns, and $\sigma_1 \geq \ldots \geq \sigma_r$. Next, for any vector $\boldsymbol{z} \in \mathbb{R}^r$, since the columns of $\boldsymbol{Y}$ have unit norm,

$$\|\boldsymbol{Y}\boldsymbol{z}\|_2 \leq \sum_{l \leq r} |z_l| \|\boldsymbol{y}_l\|_2 \leq \|\boldsymbol{z}\|_1 \leq \sqrt{r} \|\boldsymbol{z}\|_2.$$

In the last inequality we used Cauchy-Schwartz. Therefore, all singular values of $\boldsymbol{Y}$ are upper bounded by $\sigma_1 \leq \sqrt{r}$. Thus, with $r' = \lceil r/s \rceil$

$$\delta^r \leq \det(\boldsymbol{Y}) = \prod_{l=1}^{r} \sigma_l \leq r^{(r'-1)/2} \sigma_{r'}^{r-r'+1} \leq r^{r/2s} \sigma_{r'}^{(s-1)r/s},$$

so that $\sigma_{r'} \geq \delta^{s/(s-1)} / r^{1/(2s)}$. We are ready to define the new vectors. We pose for all $i \leq r'$, $\boldsymbol{z}_i = \boldsymbol{u}_i$ the i-th column of $\boldsymbol{U}$. These correspond to the r' largest singular values of $\boldsymbol{Y}$ and are orthonormal by construction. Then, for any $i \leq r'$, we also have $\boldsymbol{z}_i = \boldsymbol{u}_i = \frac{1}{\sigma_i} \boldsymbol{Y} \boldsymbol{v}_i$ where $\boldsymbol{v}_i \in \mathbb{R}^r$ is the i-th column of $\boldsymbol{V}$. Hence, for any $\boldsymbol{a} \in \mathbb{R}^d$,

$$|\boldsymbol{z}_i^\top \boldsymbol{a}| = \frac{1}{\sigma_i} |\boldsymbol{v}_i^\top \boldsymbol{Y}^\top \boldsymbol{a}| \leq \frac{\|\boldsymbol{v}_i\|_1}{\sigma_i} \|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty \leq \frac{r^{1/2+1/(2s)}}{\delta^{s/(s-1)}} \|\boldsymbol{Y}^\top \boldsymbol{a}\|_\infty.$$

This ends the proof of the lemma. ■

References

[Alg92] P. Algoet. "Universal Schemes for Prediction, Gambling and Portfolio Selection". In: *The Annals of Probability* 20.2 (1992), pp. 901–941.

[Alo+21] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. "Adversarial laws of large numbers and optimal regret in online classification". In: *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*. 2021, pp. 447–455.

[AP23] Jason M Altschuler and Pablo A Parrilo. "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule". In: *arXiv preprint arXiv:2309.07879* (2023).

[Ans97] Kurt M Anstreicher. "On Vaidya's volumetric cutting plane method for convex programming". In: *Mathematics of Operations Research* 22.1 (1997), pp. 63–89.

[Ans00] Kurt M Anstreicher. "The volumetric barrier for semidefinite programming". In: *Mathematics of Operations Research* 25.3 (2000), pp. 365–380.

[Ans98] Kurt M Anstreicher. "Towards a practical volumetric cutting plane method for convex programming". In: *SIAM Journal on Optimization* 9.1 (1998), pp. 190–206.

[AS15] Yossi Arjevani and Ohad Shamir. "Communication Complexity of Distributed Convex Learning and Optimization". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 1756–1764.

[AV95] David S Atkinson and Pravin M Vaidya. "A cutting plane algorithm for convex programming that uses analytic centers". In: *Mathematical programming* 69.1-3 (1995), pp. 1–43.

[Aue+02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The nonstochastic multiarmed bandit problem". In: *SIAM Journal on Computing* 32.1 (2002), pp. 48–77.

[AC16] Peter Auer and Chao-Kai Chiang. "An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits". In: *Conference on Learning Theory*. PMLR. 2016, pp. 116–120.

[BS18] Eric Balkanski and Yaron Singer. "Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization". In: *arXiv preprint arXiv:1808.03880* (2018).

[BOY18] Paul Beame, Shayan Oveis Gharan, and Xin Yang. "Time-Space Tradeoffs for Learning Finite Functions from Random Evaluations, with Applications to Polynomials". In: *Proceedings of the 31st Conference On Learning Theory*. PMLR, 2018, pp. 843–856.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. "Agnostic Online Learning." In: *Conference on Learning Theory*. Vol. 3. 2009, p. 1.

[BU12] Shai Ben-David and Ruth Urner. "On the hardness of domain adaptation and the utility of unlabeled target samples". In: *International Conference on Algorithmic Learning Theory*. Springer. 2012, pp. 139–153.

[BN05] Aharon Ben-Tal and Arkadi Nemirovski. "Non-euclidean restricted memory level method for large-scale convex optimization". In: *Mathematical Programming* 102.3 (Jan. 2005), pp. 407–456.

[BV04] Dimitris Bertsimas and Santosh Vempala. "Solving convex programs by random walks". In: *Journal of the ACM (JACM)* 51.4 (2004), pp. 540–556.

[BGZ14] Omar Besbes, Yonatan Gur, and Assaf Zeevi. "Stochastic multi-armed-bandit problem with non-stationary rewards". In: *Advances in neural information processing systems* 27 (2014).

[Bla24] Moïse Blanchard. "Gradient Descent is Pareto-Optimal in the Oracle Complexity and Memory Tradeoff for Feasibility Problems". In: *arXiv preprint arXiv:2404.06720* (2024).

[Bla22] Moïse Blanchard. "Universal online learning: An optimistically universal learning rule". In: *Conference on Learning Theory*. PMLR. 2022, pp. 479–495.

[BC22] Moïse Blanchard and Romain Cosson. "Universal Online Learning with Bounded Loss: Reduction to Binary Classification". In: *Conference on Learning Theory*. PMLR. 2022, pp. 479–495.

[BCH22] Moïse Blanchard, Romain Cosson, and Steve Hanneke. "Universal Online Learning with Unbounded Losses: Memory Is All You Need". In: *International Conference on Algorithmic Learning Theory*. PMLR. 2022, pp. 107–127.

[BHJ23] Moïse Blanchard, Steve Hanneke, and Patrick Jaillet. "Adversarial Rewards in Universal Learning for Contextual Bandits". In: *arXiv preprint arXiv:2302.07186* (2023).

[BHJ22] Moïse Blanchard, Steve Hanneke, and Patrick Jaillet. "Contextual Bandits and Optimistically Universal Learning". In: *arXiv preprint arXiv:2301.00241* (2022).

[BJ22] Moïse Blanchard and Adam Quinn Jaffe. "Fréchet Mean Set Estimation in the Hausdorff Metric, via Relaxation". In: *arXiv preprint arXiv:2212.12057* (2022).

[BJ23] Moïse Blanchard and Patrick Jaillet. "Universal Regression with Adversarial Responses". In: *The Annals of Statistics* 51.3 (2023), pp. 1401–1426.

[BZJ24] Moïse Blanchard, Junhui Zhang, and Patrick Jaillet. "Memory-Constrained Algorithms for Convex Optimization". In: *Advances in Neural Information Processing Systems* 36 (2024).

[BZJ23] Moïse Blanchard, Junhui Zhang, and Patrick Jaillet. "Quadratic memory is necessary for optimal query complexity in convex optimization: Center-of-mass is pareto-optimal". In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR. 2023, pp. 4696–4736.

[Blu+89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warmuth. "Learnability and the Vapnik-Chervonenkis Dimension". In: *Journal of the Association for Computing Machinery* 36.4 (1989), pp. 929–965.

[Bou+21] O. Bousquet, S. Hanneke, S. Moran, R. van Handel, and A. Yehudayoff. "A Theory of Universal Learning". In: *Proceedings of the* 53rd *Annual ACM Symposium on Theory of Computing*. 2021.

[Bra+16] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L. Nguyen, and David P. Woodruff. "Communication Lower Bounds for Statistical Estimation Problems via a Distributed Data Processing Inequality". In: *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 1011–1020.

[Bro+21] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. "When is Memorization of Irrelevant Training Data Necessary for High-Accuracy Learning?" In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2021. Virtual, Italy: Association for Computing Machinery, 2021, pp. 123–132.

[BBS22] Gavin Brown, Mark Bun, and Adam Smith. "Strong Memory Lower Bounds for Learning Natural Models". In: *Proceedings of Thirty Fifth Conference on Learning Theory*. PMLR, 2022, pp. 4989–5029.

[Bro70] C. G. Broyden. "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations". In: *IMA Journal of Applied Mathematics* 6.1 (Mar. 1970), pp. 76–90. ISSN: 0272-4960.

[Bub15] Sébastien Bubeck. "Convex Optimization: Algorithms and Complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357. ISSN: 1935-8237.

[BC+12] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122.

[Bub+19] Sébastien Bubeck, Qijia Jiang, Yin-Tat Lee, Yuanzhi Li, and Aaron Sidford. "Complexity of highly parallel non-smooth convex optimization". In: *Advances in neural information processing systems* 32 (2019).

[BLS15] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. "A geometric alternative to Nesterov's accelerated gradient descent". In: (2015). arXiv: 1506.08187 [math.OC].

[BM20] Sébastien Bubeck and Dan Mikulincer. "How to trap a gradient flow". In: *Conference on Learning Theory*. PMLR. 2020, pp. 940–960.

[CG06] Frédéric Cérou and Arnaud Guyader. "Nearest neighbor classification in infinite dimension". In: *ESAIM: Probability and Statistics* 10 (2006), pp. 340–355.

[Ces+97a] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. "How to Use Expert Advice". In: *Journal of the Association for Computing Machinery* 44.3 (1997), pp. 427–485.

[CL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[Ces+97b] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. "How to use expert advice". In: *Journal of the ACM (JACM)* 44.3 (1997), pp. 427–485.

[Cha89] Ted Chang. "Spherical regression with errors in variables". In: *The Annals of Statistics* (1989), pp. 293–306.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[CD14] Kamalika Chaudhuri and Sanjoy Dasgupta. "Rates of convergence for nearest neighbor classification". In: *Advances in Neural Information Processing Systems* 27 (2014).

[CP23] Xi Chen and Binghui Peng. "Memory-query tradeoffs for randomized convex optimization". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 1400–1413.

[Che+19] Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. "A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free". In: *Conference on Learning Theory*. PMLR. 2019, pp. 696–726.

[CK22] Dan Tsir Cohen and Aryeh Kontorovich. "Learning with metric losses". In: *Conference on Learning Theory*. PMLR. 2022, pp. 662–700.

[Col84] G. Collomb. "Propriétés de Convergence Presque Complète du Prédicteur à Noyau". In: *Zeitschrift f ur Wahrscheinlichkeitstheorie und verwandte Gebiete* 66 (1984), pp. 441–460.

[CH67] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[dST21] Alexandre d'Aspremont, Damien Scieur, and Adrien Taylor. "Acceleration Methods". In: *Foundations and Trends® in Optimization* 5.1-2 (2021), pp. 1–245.

[DKS19] Yuval Dagan, Gil Kur, and Ohad Shamir. "Space lower bounds for linear prediction in the streaming model". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, 2019, pp. 929–954.

[DS18] Yuval Dagan and Ohad Shamir. "Detecting Correlations with Little Memory and Communication". In: *Proceedings of the 31st Conference On Learning Theory*. Ed. by Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Vol. 75. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1145–1198.

[DVR23] Shuvomoy Das Gupta, Bart PG Van Parys, and Ernest K Ryu. "Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods". In: *Mathematical Programming* (2023), pp. 1–73.

[DS01] Kenneth R Davidson and Stanislaw J Szarek. "Local operator theory, random matrices and Banach spaces". In: *Handbook of the geometry of Banach spaces* 1.317-366 (2001), p. 131.

[Dav+10] Brad C Davis, P Thomas Fletcher, Elizabeth Bullitt, and Sarang Joshi. "Population shape regression from random design data". In: *International Journal of Computer Vision* 90.2 (2010), pp. 255–266.

[DD78] Carl De Boor and Carl De Boor. *A practical guide to splines*. Vol. 27. springer-verlag New York, 1978.

[DDH07] James Demmel, Ioana Dumitriu, and Olga Holtz. "Fast linear algebra is stable". In: *Numerische Mathematik* 108.1 (2007), pp. 59–91.

[Dev+94] Luc Devroye, Laszlo Gyorfi, Adam Krzyzak, and Gábor Lugosi. "On the strong universal consistency of nearest neighbor regression function estimates". In: *The Annals of Statistics* (1994), pp. 1371–1385.

[DGL13] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science & Business Media, 2013.

[DF14] Kim Donghwan and Jeffrey Fessler. "Optimized first-order methods for smooth convex minimization". In: *Mathematical Programming* 159 (June 2014).

[DT14] Yoel Drori and Marc Teboulle. "Performance of first-order methods for smooth convex minimization: a novel approach". In: *Mathematical Programming* 145.1 (2014), pp. 451–482.

[DFR18] Dmitriy Drusvyatskiy, Maryam Fazel, and Scott Roy. "An Optimal First Order Method Based on Optimal Quadratic Averaging". In: *SIAM Journal on Optimization* 28.1 (2018), pp. 251–271.

[EJ20] Steven N Evans and Adam Q Jaffe. "Strong laws of large numbers for Fréchet means". In: *arXiv preprint arXiv:2012.12859* (2020).

[FS02] Uriel Feige and Gideon Schechtman. "On the optimality of the random hyperplane rounding technique for MAX CUT". In: *Random Structures & Algorithms* 20.3 (2002), pp. 403–440.

[Fle13] P. T. Fletcher. "Geodesic regression and the theory of least squares on Riemannian manifolds". In: *International Journal of Computer Vision* 105.2 (2013), pp. 171–185.

[Fle70] R. Fletcher. "A new approach to variable metric algorithms". In: *The Computer Journal* 13.3 (Jan. 1970), pp. 317–322. ISSN: 0010-4620.

[FR64] Reeves Fletcher and Colin M Reeves. "Function minimization by conjugate gradients". In: *The computer journal* 7.2 (1964), pp. 149–154.

[FKL20] D. J. Foster, A. Krishnamurthy, and H. Luo. "Open Problem: Model Selection for Contextual Bandits". In: *Proceedings of the* 33rd *Conference on Learning Theory*. 2020.

[FS97] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[Gar+21] Sumegha Garg, Pravesh K Kothari, Pengda Liu, and Ran Raz. "Memory-sample lower bounds for learning parity with noise". In: *arXiv preprint arXiv:2107.02320* (2021).

[GRT18] Sumegha Garg, Ran Raz, and Avishay Tal. "Extractor-Based Time-Space Lower Bounds for Learning". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2018. Association for Computing Machinery, 2018, pp. 990–1002.

[GRT19] Sumegha Garg, Ran Raz, and Avishay Tal. "Time-space lower bounds for two-pass learning". In: *34th Computational Complexity Conference (CCC)*. 2019.

[GZ09] Alexander Goldenshluger and Assaf Zeevi. "Woodroofe's one-armed bandit problem revisited". In: *The Annals of Applied Probability* 19.4 (2009), pp. 1603–1633.

[Gol70] Donald Goldfarb. "A Family of Variable-Metric Methods Derived by Variational Means". In: *Mathematics of Computation* 24.109 (1970), pp. 23–26.

[GG09] Robert M Gray and RM Gray. *Probability, random processes, and ergodic properties*. Vol. 1. Springer, 2009.

[GS93] Peter J Green and Bernard W Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. Crc Press, 1993.

[Gre+09] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. "Covariate shift by kernel mean matching". In: *Dataset shift in machine learning* 3.4 (2009), p. 5.

[GLS12] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Vol. 2. Springer Science & Business Media, 2012.

[GJ18] Melody Guan and Heinrich Jiang. "Nonparametric stochastic contextual bandits". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[GL02] László Györfi and Gábor Lugosi. "Strategies for sequential prediction of stationary time series". In: *Modeling uncertainty*. Springer, 2002, pp. 225–248.

[GLM99] L Gyorfi, Gábor Lugosi, and Gusztáv Morvai. "A simple randomized algorithm for sequential prediction of ergodic time series". In: *IEEE Transactions on Information Theory* 45.7 (1999), pp. 2642–2650.

[Gyö+02] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag New York, 2002.

[GO07] László Györfi and György Ottucsák. "Sequential prediction of unbounded stationary time series". In: *IEEE Transactions on Information Theory* 53.5 (2007), pp. 1866–1872.

[GW21] László Györfi and Roi Weiss. "Universal consistency and rates of convergence of multiclass prototype algorithms in metric spaces". In: *Journal of Machine Learning Research* 22.151 (2021), pp. 1–25.

[HZ06] William W Hager and Hongchao Zhang. "A survey of nonlinear conjugate gradient methods". In: *Pacific journal of Optimization* 2.1 (2006), pp. 35–58.

[Han21a] S. Hanneke. "Learning Whenever Learning Is Possible: Universal Learning under General Stochastic Processes". In: *Journal of Machine Learning Research* 22 (2021), pp. 1–116.

[Han21b] Steve Hanneke. "Open Problem: Is There an Online Learning Algorithm That Learns Whenever Online Learning Is Possible?" In: *Conference on Learning Theory*. PMLR. 2021, pp. 4642–4646.

[Han16] Steve Hanneke. "The optimal sample complexity of PAC learning". In: *Journal of Machine Learning Research* 17.38 (2016), pp. 1–15.

[Han22] Steve Hanneke. "Universally consistent online learning with arbitrarily dependent responses". In: *International Conference on Algorithmic Learning Theory*. PMLR. 2022, pp. 488–497.

[Han+21] Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. "Universal Bayes consistency in metric spaces". In: *The Annals of Statistics* 49.4 (2021), pp. 2129–2150.

[HW71] David Lee Hanson and Farroll Tim Wright. "A bound on tail probabilities for quadratic forms in independent random variables". In: *The Annals of Mathematical Statistics* 42.3 (1971), pp. 1079–1083.

[HMB15] Negar Hariri, Bamshad Mobasher, and Robin Burke. "Adapting to user preference changes in interactive recommendation". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[HLW94] D. Haussler, N. Littlestone, and M. Warmuth. "Predicting $\{0,1\}$-Functions on Randomly Drawn Points". In: *Information and Computation* 115.2 (1994), pp. 248–292.

[Irl97] A. Irle. "On Consistency in Nonparametric Estimation under Mixing Conditions". In: *Journal of Multivariate Analysis* 60.1 (1997), pp. 123–147.

[Jaf22] Adam Quinn Jaffe. "Strong consistency for a class of adaptive clustering procedures". In: *arXiv preprint arXiv:2202.13423* (2022).

[JL11] Albert Xin Jiang and Kevin Leyton-Brown. "Polynomial-time computation of exact correlated equilibrium in compact games". In: *Proceedings of the 12th ACM conference on Electronic commerce*. 2011, pp. 119–126.

[Jia21] Haotian Jiang. "Minimizing convex functions with integral minimizers". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2021, pp. 976–985.

[Jia+20] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. "An improved cutting plane method for convex optimization, convex-concave games, and its applications". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 944–953.

[KA16] Zohar S Karnin and Oren Anava. "Multi-armed bandits: Competing with optimal sequences". In: *Advances in Neural Information Processing Systems* 29 (2016).

[Kec12] Alexander Kechris. *Classical descriptive set theory*. Vol. 156. Springer Science & Business Media, 2012.

[Kel+22] Jonathan Kelner, Annie Marsden, Vatsal Sharan, Aaron Sidford, Gregory Valiant, and Honglin Yuan. "Big-Step-Little-Step: Efficient Gradient Methods for Objectives with Multiple Scales". In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Ed. by Po-Ling Loh and Maxim Raginsky. Vol. 178. Proceedings of Machine Learning Research. PMLR, 2022, pp. 2431–2540.

[Kha80] Leonid G Khachiyan. "Polynomial algorithms in linear programming". In: *USSR Computational Mathematics and Mathematical Physics* 20.1 (1980), pp. 53–72.

[KRT17] Gillat Kol, Ran Raz, and Avishay Tal. "Time-Space Hardness of Learning Sparse Parities". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2017. Association for Computing Machinery, 2017, pp. 1067–1080.

[Lan15] Guanghui Lan. "Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization". In: *Mathematical Programming* 149.1 (Feb. 2015), pp. 1–45.

[LLZ20] Guanghui Lan, Soomin Lee, and Yi Zhou. "Communication-efficient algorithms for decentralized and stochastic optimization". In: *Mathematical Programming* 180.1 (Mar. 2020), pp. 237–284.

[LZ07] John Langford and Tong Zhang. "The epoch-greedy algorithm for multi-armed bandits with side information". In: *Advances in neural information processing systems* 20 (2007).

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[LM00] Beatrice Laurent and Pascal Massart. "Adaptive estimation of a quadratic functional by model selection". In: *Annals of statistics* (2000), pp. 1302–1338.

[LSV07] Quoc Le, Alex Smola, and S.V.N. Vishwanathan. "Bundle Methods for Machine Learning". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2007.

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. "A faster cutting plane method and its implications for combinatorial and convex optimization". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 1049–1065.

[LNN95] Claude Lemaréchal, Arkadi Nemirovski, and Yurii Nesterov. "New variants of bundle methods". In: *Mathematical Programming* 69.1 (July 1995), pp. 111–147.

[Lev65] Anatoly Yur'evich Levin. "An algorithm for minimizing convex functions". In: *Doklady Akademii Nauk*. Vol. 160. 6. Russian Academy of Sciences. 1965, pp. 1244–1247.

[LO13] Adrian S. Lewis and Michael L. Overton. "Nonsmooth optimization via quasi-Newton methods". In: *Mathematical Programming* 141.1 (Oct. 2013), pp. 135–163.

[LM21] Zhenhua Lin and Hans-Georg Müller. "Total variation regularized Fréchet regression for metric-space valued data". In: *The Annals of Statistics* 49.6 (2021), pp. 3510–3533.

[Lit88] N. Littlestone. "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm". In: *Machine Learning* 2.4 (1988), pp. 285–318.

[LW94] Nick Littlestone and Manfred K Warmuth. "The weighted majority algorithm". In: *Information and Computation* 108.2 (1994), pp. 212–261.

[LN89] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1 (Aug. 1989), pp. 503–528.

[LLS18] Fang Liu, Joohyun Lee, and Ness Shroff. "A change-detection based framework for piecewise-stationary multi-armed bandit problem". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[LKS06] A. C. Lozano, S. R. Kulkarni, and R. E. Schapire. "Convergence and Consistency of Regularized Boosting Algorithms with Stationary β-mixing Observations". In: *Advances in Neural Information Processing Systems 18*. 2006.

[LPP09] Tyler Lu, Dávid Pál, and Martin Pál. "Showing relevant ads via context multi-armed bandits". In: *Proceedings of AISTATS*. 2009.

[Luo+18] Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. "Efficient contextual bandits in non-stationary worlds". In: *Conference On Learning Theory*. PMLR. 2018, pp. 1739–1776.

[Mal99] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[MJM00] Kanti V Mardia, Peter E Jupp, and KV Mardia. *Directional statistics*. Vol. 2. Wiley Online Library, 2000.

[MZ21] T. Marinov and J. Zimmert. "The Pareto Frontier of Model Selection for General Contextual Bandits". In: *Advances in Neural Information Processing Systems 34*. 2021.

[Mar+22] Annie Marsden, Vatsal Sharan, Aaron Sidford, and Gregory Valiant. "Efficient convex optimization requires superlinear memory". In: *Conference on Learning Theory*. PMLR. 2022, pp. 2390–2430.

[McC05] S Thomas McCormick. "Submodular function minimization". In: *Handbooks in operations research and management science* 12 (2005), pp. 321–391.

[MCJ13] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. "Memory Limited, Streaming PCA". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2886–2894.

[MKN99] Gusztáv Morvai, Sanjeev R Kulkarni, and Andrew B Nobel. "Regression estimation from an individual stable sequence". In: *Statistics: A Journal of Theoretical and Applied Statistics* 33.2 (1999), pp. 99–118.

[MYG96] Gusztáv Morvai, Sidney Yakowitz, and László Györfi. "Nonparametric inference for ergodic, stationary time series". In: *The Annals of Statistics* 24.1 (1996), pp. 370–379.

[MM18] Dana Moshkovitz and Michal Moshkovitz. "Entropy Samplers and Strong Generic Lower Bounds For Space Bounded Learning". In: *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Vol. 94. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 28:1–28:20.

[MM17] Dana Moshkovitz and Michal Moshkovitz. "Mixing Implies Lower Bounds for Space Bounded Learning". In: *Proceedings of the 2017 Conference on Learning Theory*. PMLR, 2017, pp. 1516–1566.

[Mot+13] João F. C. Mota, João M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. "D-ADMM: A Communication-Efficient Distributed Algorithm for Separable Optimization". In: *IEEE Transactions on Signal Processing* 61.10 (2013), pp. 2718–2723.

[Nem94] Arkadi Nemirovski. "On parallel complexity of nonsmooth convex optimization". In: *Journal of Complexity* 10.4 (1994), pp. 451–463.

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. "Problem complexity and method efficiency in optimization". In: (1983).

[NYD83] A.S. Nemirovsky, D.B. Yudin, and E.R. Dawson. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983. ISBN: 978-0471103455.

[Nes89] Ju E Nesterov. "Self-concordant functions and polynomial-time methods in convex programming". In: *Report, Central Economic and Mathematic Institute, USSR Acad. Sci* (1989).

[Nes83] Yurii Nesterov. "A method of solving a convex programming problem with convergence rate $O(1/k^2)$". In: *Dokl. Akad. Nauk SSSR* 269 (3 1983), pp. 543–547.

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2003.

[Neu15] Gergely Neu. "Explore no more: Improved high-probability regret bounds for non-stochastic bandits". In: *Advances in Neural Information Processing Systems* 28 (2015).

[Nob03] A. B. Nobel. "On Optimal Sequential Prediction for General Processes". In: *IEEE Transactions on Information Theory* 49.1 (2003), pp. 83–98.

[Noc80] Jorge Nocedal. "Updating Quasi-Newton Matrices with Limited Storage". In: *Mathematics of Computation* 35.151 (1980), pp. 773–782. (Visited on 02/04/2023).

[Orn78] D. S. Ornstein. "Guessing the Next Output of a Stationary Process". In: *Israel Journal of Mathematics* 30.3 (1978), pp. 292–296.

[PR08] Christos H Papadimitriou and Tim Roughgarden. "Computing correlated equilibria in multi-player games". In: *Journal of the ACM (JACM)* 55.3 (2008), pp. 1–29.

[PR23] Binghui Peng and Aviad Rubinstein. "Near optimal memory-regret tradeoff for online learning". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 1171–1194.

[PZ23] Binghui Peng and Fred Zhang. "Online prediction in sub-linear space". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2023, pp. 1611–1634.

[PR13] Vianney Perchet and Philippe Rigollet. "The multi-armed bandit problem with covariates". In: *The Annals of Statistics* 41.2 (2013), pp. 693–721.

[PW17] Mert Pilanci and Martin J Wainwright. "Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence". In: *SIAM Journal on Optimization* 27.1 (2017), pp. 205–245.

[RST15a] A. Rakhlin, K. Sridharan, and A. Tewari. "Online Learning via Sequential Complexities". In: *Journal of Machine Learning Research* 16.2 (2015), pp. 155–186.

[RS16] Alexander Rakhlin and Karthik Sridharan. "Bistro: An efficient relaxation-based method for contextual bandits". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1977–1985.

[RST15b] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. "Online learning via sequential complexities." In: *J. Mach. Learn. Res.* 16.1 (2015), pp. 155–186.

[RM95] Srinivasan Ramaswamy and John E Mitchell. *A long step cutting plane algorithm that uses the volumetric barrier*. Tech. rep. Citeseer, 1995.

[Raz17] Ran Raz. "A Time-Space Lower Bound for a Large Class of Learning Problems". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. 2017, pp. 732–742.

[Raz18] Ran Raz. "Fast learning requires good memory: A time-space lower bound for parity learning". In: *Journal of the ACM (JACM)* 66.1 (2018), pp. 1–18.

[Red+16] Sashank J. Reddi, Jakub Konecný, Peter Richtárik, Barnabás Póczos, and Alex Smola. "AIDE: Fast and Communication Efficient Distributed Optimization". In: *ArXiv* abs/1608.06879 (2016).

[RMB18] Henry Reeve, Joe Mellor, and Gavin Brown. "The k-nearest neighbour ucb algorithm for multi-armed bandits with covariates". In: *Algorithmic Learning Theory*. PMLR. 2018, pp. 725–752.

[RZ10] Philippe Rigollet and Assaf Zeevi. "Nonparametric bandits with covariates". In: *arXiv preprint arXiv:1003.1630* (2010).

[RM19] Farbod Roosta-Khorasani and Michael W Mahoney. "Sub-sampled Newton methods". In: *Mathematical Programming* 174 (2019), pp. 293–326.

[Rou88] G. G. Roussas. "Nonparametric Estimation in Mixing Sequences of Random Variables". In: *Journal of Statistical Planning and Inference* 18 (1988), pp. 135–149.

[RV13] Mark Rudelson and Roman Vershynin. "Hanson-Wright inequality and sub-Gaussian concentration". In: (2013).

[RZ16] Mark Rudelson and Ofer Zeitouni. "Singular values of Gaussian matrices and permanent estimators". In: *Random Structures & Algorithms* 48.1 (2016), pp. 183–212.

[Rya06] D. Ryabko. "Pattern Recognition for Conditionally Independent Data". In: *Journal of Machine Learning Research* 7.4 (2006), pp. 645–664.

[Sar91] Jyotirmoy Sarkar. "One-armed bandit problems with covariates". In: *The Annals of Statistics* (1991), pp. 1978–2002.

[SS02] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[Sch22] Christof Schötz. "Strong laws of large numbers for generalizations of Fréchet mean sets". In: *Statistics* (2022), pp. 1–19.

[SFL17] B. Van Scoy, R. A. Freeman, and K. M. Lynch. "The Fastest Known Globally Convergent First-Order Method for Minimizing Strongly Convex Functions". In: *IEEE Control Systems Letters* PP.99 (2017), pp. 1–1.

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[SSZ14] Ohad Shamir, Nati Srebro, and Tong Zhang. "Communication-Efficient Distributed Optimization using an Approximate Newton-type Method". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1000–1008.

[Sha70] D. F. Shanno. "Conditioning of Quasi-Newton Methods for Function Minimization". In: *Mathematics of Computation* 24.111 (1970), pp. 647–656. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2004840 (visited on 05/10/2023).

[SSV19] Vatsal Sharan, Aaron Sidford, and Gregory Valiant. "Memory-Sample Tradeoffs for Linear Regression with Small Error". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. Association for Computing Machinery, 2019, pp. 890–901.

[Shi+09] Xiaoyan Shi, Martin Styner, Jeffrey Lieberman, Joseph G Ibrahim, Weili Lin, and Hongtu Zhu. "Intrinsic regression models for manifold-valued data". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2009, pp. 192–199.

[Sho77] Naum Z Shor. "Cut-off method with space extension in convex programming problems". In: *Cybernetics* 13.1 (1977), pp. 94–96.

[Sli11] Aleksandrs Slivkins. "Contextual bandits with similarity information". In: *Proceedings of the 24th annual Conference On Learning Theory*. JMLR Workshop and Conference Proceedings. 2011, pp. 679–702.

[Sli+19] Aleksandrs Slivkins et al. "Introduction to multi-armed bandits". In: *Foundations and Trends® in Machine Learning* 12.1-2 (2019), pp. 1–286.

[Smi+17] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takáč, Michael I. Jordan, and Martin Jaggi. "CoCoA: A General Framework for Communication-Efficient Distributed Optimization". In: *J. Mach. Learn. Res.* 18.1 (2017), pp. 8590–8638. ISSN: 1532-4435.

[Sri+22] Vaidehi Srinivas, David P Woodruff, Ziyu Xu, and Samson Zhou. "Memory bounds for the experts problem". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 2022, pp. 1158–1171.

[SD15] Jacob Steinhardt and John Duchi. "Minimax rates for memory-bounded sparse linear regression". In: *Proceedings of The 28th Conference on Learning Theory*. PMLR, 2015, pp. 1564–1587.

[SVW16] Jacob Steinhardt, Gregory Valiant, and Stefan Wager. "Memory, Communication, and Statistical Queries". In: *29th Annual Conference on Learning Theory*. PMLR, 2016, pp. 1490–1516.

[SHS09] Ingo Steinwart, Don Hush, and Clint Scovel. "Learning from dependent observations". In: *Journal of Multivariate Analysis* 100.1 (2009), pp. 175–194.

[Ste85] Jacques Stern. "Le probleme de la mesure". In: *Séminaire Bourbaki* 1983 (1985), p. 84.

[Sto77] Charles J Stone. "Consistent nonparametric regression". In: *The Annals of Statistics* (1977), pp. 595–620.

[Sug+07] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. "Direct importance estimation with model selection and its application to covariate shift adaptation". In: *Advances in neural information processing systems* 20 (2007).

[SK21] Joseph Suk and Samory Kpotufe. "Self-tuning bandits over unknown covariate-shifts". In: *Algorithmic Learning Theory*. PMLR. 2021, pp. 1114–1156.

[Tao23] Terence Tao. *Topics in random matrix theory*. Vol. 132. American Mathematical Society, 2023.

[Tar88] Sergei Pavlovich Tarasov. "The method of inscribed ellipsoids". In: *Soviet Mathematics-Doklady*. Vol. 37. 1. 1988, pp. 226–230.

[TD23] Adrien Taylor and Yoel Drori. "An optimal gradient method for smooth strongly convex minimization". In: *Mathematical Programming* 199.1 (May 2023), pp. 557–594.

[Teo+10] Choon Hui Teo, S. V. N. Vishwanthan, Alex J. Smola, and Quoc V. Le. "Bundle Methods for Regularized Risk Minimization". In: *Journal of Machine Learning Research* 11.10 (2010), pp. 311–365.

[TK22] Dan Tsir Cohen and Aryeh Kontorovich. "Metric-valued regression". In: *Submitted to COLT* (2022).

[Tsy09] Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. springer, 2009.

[UB13] R. Urner and S. Ben-David. "Probabilistic Lipschitzness A Niceness Assumption for Deterministic Labels". In: *Learning Faster from Easy Data-Workshop @ NIPS*. 2013.

[Vai96] Pravin M Vaidya. "A new algorithm for minimizing convex functions over convex sets". In: *Mathematical programming* 73.3 (1996), pp. 291–341.

[Val84] L. G. Valiant. "A Theory of the Learnable". In: *CACM* 27.11 (Nov. 1984), pp. 1134–1142.

[Vap82] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag New York, 1982.

[VC71] Vladmir Vapnik and Alexey Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities". In: *Theory of Probability and its Applications* 16 (1971), pp. 264–280.

[Vav93] Stephen A Vavasis. "Black-box complexity of local minimization". In: *SIAM Journal on Optimization* 3.1 (1993), pp. 60–80.

[Ver20] Roman Vershynin. "High-dimensional probability". In: *University of California, Irvine* 10 (2020), p. 11.

[VT98] Divakar Viswanath and LN Trefethen. "Condition numbers of random triangular matrices". In: *SIAM Journal on Matrix Analysis and Applications* 19.2 (1998), pp. 564–581.

[Vit05] Giuseppe Vitali. *Sul problema della misura dei Gruppi di punti di una retta: Nota*. Tip. Gamberini e Parmeggiani, 1905.

[Wah90] Grace Wahba. *Spline models for observational data*. SIAM, 1990.

[WKP05] Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. "Bandit problems with side observations". In: *IEEE Transactions on Automatic Control* 50.3 (2005), pp. 338–355.

[WWS17] Jialei Wang, Weiran Wang, and Nathan Srebro. "Memory and Communication Efficient Distributed Stochastic Optimization with Minibatch Prox". In: *Proceedings of the 2017 Conference on Learning Theory*. Ed. by Satyen Kale and Ohad Shamir. Vol. 65. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1882–1919.

[Wan+18] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. "Gradient Sparsification for Communication-Efficient Distributed Optimization". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 1306–1316.

[Was06] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[Woo79] Michael Woodroofe. "A one-armed bandit problem with a concomitant variable". In: *Journal of the American Statistical Association* 74.368 (1979), pp. 799–806.

[WS19] Blake Woodworth and Nathan Srebro. "Open problem: The oracle complexity of convex optimization with limited memory". In: *Conference on Learning Theory*. PMLR. 2019, pp. 3202–3210.

[Woo+21] Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. "The Min-Max Complexity of Distributed Stochastic Convex Optimization with Intermittent Communication". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Ed. by Mikhail Belkin and Samory Kpotufe. Vol. 134. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4386–4437.

[WS16] Blake E Woodworth and Nati Srebro. "Tight Complexity Bounds for Optimizing Composite Objectives". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[WS17] Blake E. Woodworth and Nathan Srebro. "Lower Bound for Randomized First Order Convex Optimization". In: *arXiv: Optimization and Control* (2017).

[WIW18] Qingyun Wu, Naveen Iyer, and Hongning Wang. "Learning contextual bandits in a non-stationary environment". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 495–504.

[YZ02] Yuhong Yang and Dan Zhu. "Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates". In: *The Annals of Statistics* 30.1 (2002), pp. 100–121.

[YN76a] David Yudin and Arkadii Nemirovski. "Evaluation of the information complexity of mathematical programming problems". In: *Ekonomika i Matematicheskie Metody* 12 (1976), pp. 128–142.

[YN76b] David B Yudin and Arkadi S Nemirovskii. "Informational complexity and efficient methods for the solution of convex extremal problems". In: *Matekon* 13.2 (1976), pp. 22–45.

[Zai+19] Hanan Zaichyk, Armin Biess, Aryeh Kontorovich, and Yury Makarychev. "Efficient Kirszbraun extension with applications to regression". In: *arXiv preprint arXiv:1905.11930* (2019).

[ZDW12] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. "Communication-efficient algorithms for statistical optimization". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. 2012, pp. 6792–6792.