Near-Optimal Learning in Sequential Games

By

Tiancheng Yu

B.E., Tsinghua University (2018)
M.S. Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Tiancheng Yu
Department of Electrical Engineering and Computer Science
May 19, 2023

Certified by: Suvrit Sra
Esther and Harold E. Edgerton Career Development Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Near-Optimal Learning in Sequential Games

by

Tiancheng Yu

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Decision making is ubiquitous, and some problems become particularly challenging due to their sequential nature, where later decisions depend on earlier ones. While humans have been attempting to solve sequential decision making problems for a long time, modern computational and machine learning techniques are needed to find the optimal decision rule. One popular approach is the reinforcement learning (RL) perspective, in which an agent learns the optimal decision rule by receiving rewards based on its actions.

In the presence of multiple learning agents, sequential decision making problems become sequential games. In this setting, the learning objective shifts from finding an optimal decision rule to finding a Nash equilibrium, where none of the agents can increase their reward by unilaterally switching to another decision rule. To handle both the sequential nature of the problem and the presence of the other learning agents, multi-agent RL tasks require even more data than supervised learning and single-agent RL tasks. Consequently, sample efficiency becomes a critical concern for the success of multi-agent RL.

In this thesis, I study argubly the most fundamental problems of learning in sequential games:

1. (**Lower bound**) How many samples are necessary to find a Nash equilibrium in a sequential game, no matter what learning algorithm is used?

2. (**Upper bound**) How to design (computationally) efficient learning algorithms with sharp sample complexity guarantees?

When the upper and lower bounds match each other, (minimax) optimal learning is achieved. It turns out utilizing structures of sequential games is the key towards optimal learning. In this thesis, we investigate near-optimal learning in two types of sequential games:

1. (**Markov games**) All the agents can observe the underlying states (Chapter 2) and,

2. (**Extensive-form games**) Different agents can have different observations given the same state (Chapter 5).

To achieve near-optimal learning, a series of novel algorithmic idea and analytical tools will be introduced, such as

1. (**Adaptive uncertainty quantification**) Sharp uncertainty quantification of the value function estimations to design near-optimal exploration bonus (Chapter 3),

2. (**Certified policy**) A non-uniform and step-wise reweighting of historical policies to produce approximate Nash equilibrium policies (Chapter 4),

3. (**Balanced exploration**) Achieing optimal exploration of a game tree based on the size of the subtrees (Chapter 6),

4. (**Log-partition function reformulation**) Re-interpreting classical algorithms as computing gradients of a log-partition function (Chapter 7),

which may be of independent interest.

Thesis Supervisor: Suvrit Sra
Title: Associate Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Constantinos Daskalakis
Title: Avanessians Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Chi Jin
Title: Assistant Professor of Electrical and Computer Engineering
Princeton University

Acknowledgments

First of all, I'm deeply grateful to be supported by the following funding sources during my PhD years: EECS Jae S. Lim Fellowship, NSF BIGDATA grant (6937242), NSF CAREER grant (6940395) and NSF CCF-2112665 (TILOS AI Research Institute).

I would like to take this opportunity to express my deep gratitude to my committee members for their invaluable support and guidance throughout the thesis process. Their constructive feedback and insightful comments have been instrumental in shaping the structure of this thesis, and I am truly grateful for their time and effort.

I feel incredibly fortunate to have had Suvrit as my advisor. From the moment I joined his group meeting during the visit days five years ago, I knew that I had found the perfect academic home. Although we no longer have free breakfast during group meetings, the atmosphere remains as inspiring as ever. Suvrit has created a culture where everyone is both humble and ambitious, and I have greatly benefited from the diverse perspectives of my fellow group members. Through countless discussions and collaborations, I have grown as a researcher and as a person. I am grateful for the many ways in which these interactions have shaped my research interests and even my life.

Throughout my PhD journey, my appreciation for Suvrit has grown monotonically. He has been a generous and supportive mentor from the very beginning, providing me with opportunities that have shaped my research trajectory. For example, Suvrit supported my visit to Princeton, where I had the chance to work with Chi and later meet other brilliant co-authors from MSR and the west coast. Beyond his support, I have been impressed by Suvrit's sense of humor and his unwavering perseverance in the face of challenges. As I reflect on our time together, I realize that Suvrit is exactly the kind of person I aspire to become in the future.

Chi has been my most important collaborator, and I am deeply grateful for his influence on my research trajectory. It was through his paper [Jin et al., 2018] that I first became interested in RL theory, and it was his suggestion to explore game theory that truly sparked my fascination with this research topic. Although I was initially skeptical about the plan, I soon realized that it was one of the best decisions I have ever made. Through our collaboration, I have gained a deeper understanding of the field and developed new skills that have been instrumental in my research. I am grateful for Chi's guidance and friendship, and I look forward to continued collaborations in the future.

I feel incredibly fortunate that Costis agreed to join my committee, and I am grateful for his insights and comments on my research. It has been a fantastic opportunity to learn from such an accomplished researcher, and I appreciate his willingness to share his expertise with me.

Half of my PhD journey has taken place during the pandemic, which has posed unique challenges. However, I am lucky to have had the support of superb collaborators who have been instrumental in helping me navigate this difficult time. Some of the most important lessons I have learned during my PhD journey have come from these brilliant minds. Beyond their academic contributions, these individuals are also fantastic people who have made our conversations engaging and enlightening. Among my collaborators, there are a few who have been particularly important to me, and I would like to acknowledge them separately for their contributions to my research and personal growth.

I am grateful to have had Prof. Yuan Shen as my undergraduate thesis supervisor. As one of the first few undergraduate students to work in his lab, I was privileged to receive a great deal of guidance from him. He played an instrumental role in helping me to find my passion for research topics, and his mentorship was invaluable in teaching me how to write and present my research. His famous quote is

> You spend 1/3 time on research, 1/3 on writing and 1/3 on presenting.

Prof. Yanjun Han has been a role model for me since my first year in college. His most important inspiration to me is the idea that studying in an engineering school does not restrict one from becoming a skilled theoretician or mathematician. His ability to bridge the gap between intuition and implementation has always impressed me, and his work has shown me the value of interdisciplinary research.

During my visit to Stanford in 2017 summer, Prof. Jiantao Jiao served as my mentor and introduced me to the fascinating world of high-dimensional statistics. Through our collaboration, I discovered how elegant tools from different disciplines can be applied to problems that initially appear unrelated. Working with Jiantao was an eye-opening experience that taught me the importance of interdisciplinary research and showed me how to approach complex problems with both creativity and rigor.

I first met Dr. Yu Bai during my visit to Princeton in 2019, and little did I know that our collaboration would extend far beyond that initial meeting. Some of the most brilliant ideas I have had during my PhD journey were the result of our late-night discussions. I feel previledged to have had the opportunity to work with such a talented and dedicated collaborator.

I had the pleasure of collaborating with Qinghua Liu, a fellow PhD student, since our research project on reward-free learning in Markov games in 2020. Although we had known each other since our undergraduate days, this was the first time we had the opportunity to work together on a project. I was impressed by Qinghua's talent for understanding complicated and unstructured concepts and producing simple and elegant results. He has become my most reliable collaborator, and every joint project with him has been an unforgettable experience. I am confident that Qinghua will become a superstar in whatever area he chooses to work on in the future.

Brabeeba is the most inspiring and thought-provoking person I have ever met. His eagerness to learn, explore, and understand is infectious, and he is more than generous to share his ideas and energy. I consider myself fortunate to have such a friend and mentor who embodies many of the most valuable human qualities that I cherish.

Although I only collaborated with Chi-Ning Chou on one paper, we've recycled it more times than all of my other papers combined. Chi-Ning's unwavering passion and confidence in the project is the main reason we haven't given up yet, and he's always willing to lead revisions and brainstorm new ideas for potential venues. It's been amazing to see how the original draft has evolved over the years through this endeavor.

Chi-Ning's attitude towards research and life is truly inspiring, as he remains unfazed by setbacks and uncertainties. He's shown me the ideal approach to facing challenges and pursuing one's passions.

Yi Tian and Prof. Jingzhao Zhang have been invaluable colleagues in my PhD journey. Being in the same office with them has allowed us to develop a close working relationship, and they have become my go-to collaborators whenever I need to discuss anything related to control or optimization. What I admire most about them is their research style of deriving important theoretical questions from empirical experiments, which is a less popular approach in the theory community.

I also want to acknowledge my collaboration with Prof. Kaiqing Zhang and Mingyang Liu, which unfortunately started relatively late in my PhD. I regret not having started working with them earlier. The most impressive thing about Kaiqing is his ability to cover a vast amount of literature, spanning different domains and subjects. It's amazing to see how he can quickly grasp the key insights of a paper and connect them to other works in seemingly unrelated fields. His breadth of knowledge has been invaluable to our collaborations, and I feel very fortunate to have had the opportunity to work with him.

In addition to academic research, I had the opportunity to do two internships during my PhD which were incredibly helpful in broadening my horizons on the industry side. I would like to express my gratitude to my brilliant mentors, Hongyi Zhang at Bytedance and Yifei Ma at AWS AI, for their patient guidance in helping a newbie like me navigate online auction and recommender systems. I would also like to thank my colleagues Dr. Fei Feng and Dr. Ge Liu for introducing me to the AWS AI personalized team. Their support and encouragement helped make these experiences both rewarding and unforgettable.

In addition to my co-authors, I would also like to express my appreciation to the many individuals who have supported me throughout my academic journey. Their encouragement, advice, and assistance have been invaluable. Their contributions have made a significant difference, and I am honored to have had them in my corner.

When I first came to MIT, I was fortunate to receive many helpful suggestions from more senior PhDs, such as Chengtao Li, Matthew Staib, Lijie Chen, Xijia Zheng and Zhi Xu. I also gained a lot of inspiration from my cohorts, especially Yunzong Xu, Renbo Zhao, Xiyu Zhai, Liangyuan Na, Jason Liang, Yichen Yang, and Ji Lin. Additionally, I received a great deal of support from the MITCSSA community, of which I am so happy to have become a part. I would like to express my gratitude to all of my friends there, especially Zhuoran Han, Hongyin Luo, Lei Xu, Zhichu Ren, and Xinyi Gu.

During my five years at MIT, I had the privilege of living in Sidney Pacific graduate residence hall. Throughout my stay, I was fortunate to have three wonderful roommates, Baichuan Mo, Pingchuan Ma, and Yung-Sung Chuang, who have become an integral part of my life. I am grateful for their company and support, and it's saddening to bid them farewell as I move on to the next chapter of my life.

Finally, I want to express my deepest appreciation to my family. None of them have any experience in research, but they have supported my PhD journey in every way possible. Due to external factors beyond our control, we have been unable to see each other for nearly five years. However, despite the distance, I have found myself growing closer to them than ever before. I have come to appreciate their wisdom and mindset, and have been surprised to discover how similar I have become to my parents. Their unwavering support and love have been a constant source of strength and motivation throughout my PhD journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Decision-making is everywhere, ranging from choosing a life partner, trading stocks, to driving a car. The decision maker, or agent, needs to take specific actions sequentially to maximize her reward. Most of the time, the agent also needs to leverage contextual information. Decision-making in modern society has become more and more challenging as the complexity of candidate options, contextual observations, sequential structures, and, most importantly, the underlying dynamics that govern the decision-making process evolve rapidly. Therefore, the traditional approach to decision making, systematically modeling the problem and then solving it, is no longer possible.

A prevailing trend to handle such complexity is learning a decision-making model through data, i.e., the reinforcement learning (RL) approach. In a standard RL setting, the agent tries to learn a rule (the policy) to execute different actions under each different context (the state), *only* using the payoff (the reward) as feedback. This is a sharp contrast from the the conventional rule-based approach, where human experts hand craft decision rules based on their understanding of the task. See a comparison of rule-based and end-to-end decision making in Figure 1-1.

As a significant distinction from supervised learning, data is no longer drawn independently from a fixed distribution but collected following the policy. Therefore, an efficient RL algorithm requires the agent not only to maximize the reward (exploitation) but also to collect data from unvisited states (exploration). This is especially

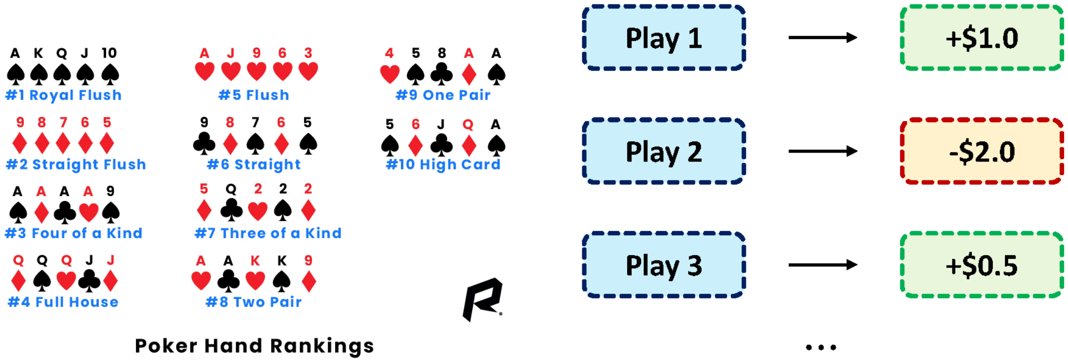


Figure 1-1: Reinforcement learning takes an end-to-end (right) instead of rule-based (left) decision making approach.

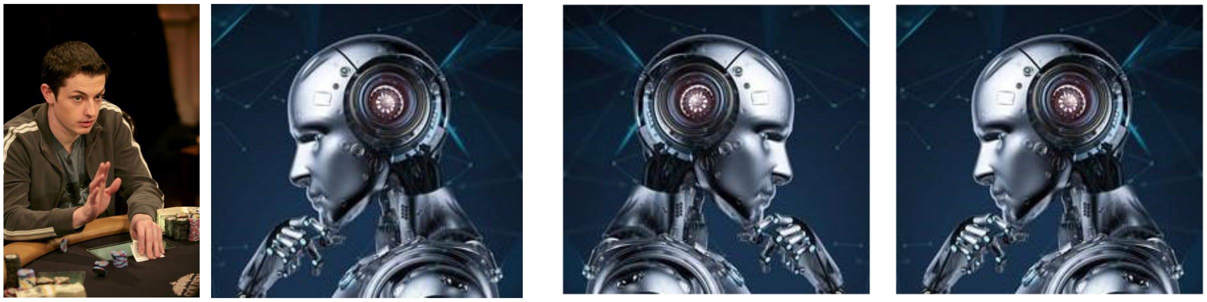


Figure 1-2: Self-play (right) v.s. playing against human expert (left).

the case when the RL agnets learn an decision rule by playing with each other instead of playing against human experts. See a comparison between self-play and playing against human expert in Figure 1-2.

The quality and diversity of the drawn data directly influence the performance of the policy learned by RL. Thus, strategic exploration is essential for RL. The influence of exploration is even more significant for complicated tasks, to which an extensive line of literature is committed. Unlike supervised learning, the agent is sometimes encouraged to visit states that have been rarely visited before, even if this will temporarily lead to a lower reward.

On the other hand, strategic exploration also makes the overall procedure sophisticated and vulnerable to non-stationarity because the agent may get confused about whether the current low reward is due to exploration or data poisoning and environmental change. See Figure 1-3 for an illustration.

Concretely, two leading factors that contribute to the non-stationrity are:

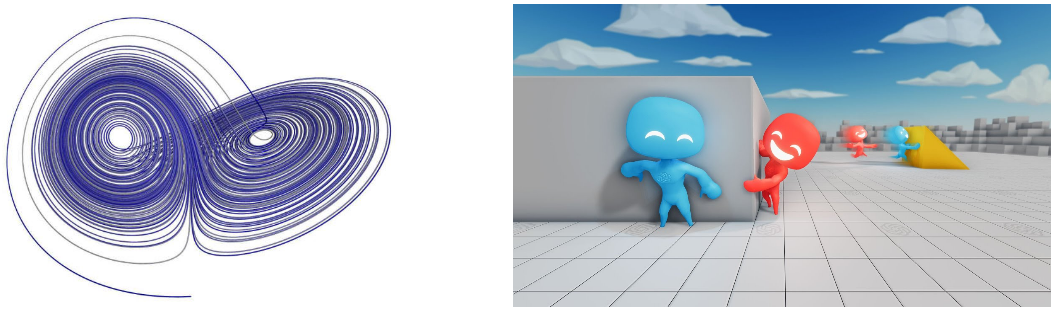


Figure 1-3: Two factors that contribute to the non-stationarity of the decision making process: non-stationary environments (left) and strategic opponents (right).

- (**Non-stationary Environments**) When the environment is non-stationary, the performance of RL algorithms can deteriorate quickly. In an otherwise stationary environment, even if the reward is perturbed slightly, popular off-the-shelf RL algorithms can be highly sub-optimal. Such non-stationarity is ubiquitous, ranging from adaptive reward engineering, malicious manipulation, simulation-to-real transition, or dataset poisoning. This issue is studied in my master thesis [Jin et al., 2020a,b] by modeling the non-stationarity explicitly.

- (**Strategic Opponents**) Modern decision-making scenarios often involve more than one agent. Even though the environment does not change, the other agents can adapt their policies consistently. This can happen in either cooperative tasks, where joint coordination is tricky in general, or competitive cases, where players' utility has confliction with each other, and every player wants to exploit the others. This topic will be the focus of this PhD thesis, by considering learning in sequential games, instead of a single-agent sequential decision making problem.

Handling non-stationarity properly requires the agent to perform (near-) optimally, even in the presence of other strategic agents. To guarantee the safety and adaptivity properties crucial in practice, we need to achieve optimal learning in such challenging conditions.

Compared with a single-agent sequential decision making problem, learning in sequential games involves new formulations and challenges beyond standard RL, and

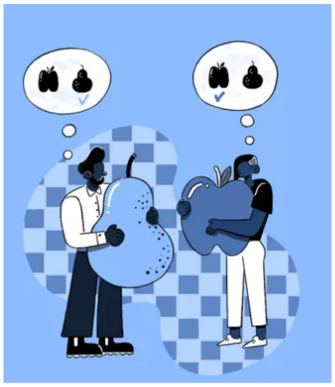


Figure 1-4: Nash equilibrium: no agent can improve her own utility by uni-laterally changing her own policy.

thus requires new mathematical and algorithmic tools.

For example, the standard notion of optimality no longer helps because the performance of an agent depends on both her own strategy and the other agents' strategy. As a result, we will use the notion of **equilibrium** instead. Informally speaking, a strategy profile of all the players constitutes a Nash equilibrium if no agent can improve her own utility by uni-laterally changing her policy. The key learning objective in sequential games is to find a Nash equilibrium.

Just like the other machine learning paradigms, RL agents are hungry for data. To measure the amount of data required to perform near-optimally, we use *sample complexity* to quantify the number of samples required to guarantee certain performance criteria. As a result, minimizing sample complexity is arguably one of the most fundamental problems behind the success of RL. This thesis aims to understand this problem:

What is the fundamental limit of sample complexity of finding an equilibrium in sequential games, and how to achieve it using computationally efficient algorithms?

When the fundamental limit is achieved by certain algorithm, we know the learning algorithm cannot be improved anymore. We call the corresponding learning algorithm a **minimax optimal learning** algorithm. In this thesis, we study minimax optimal

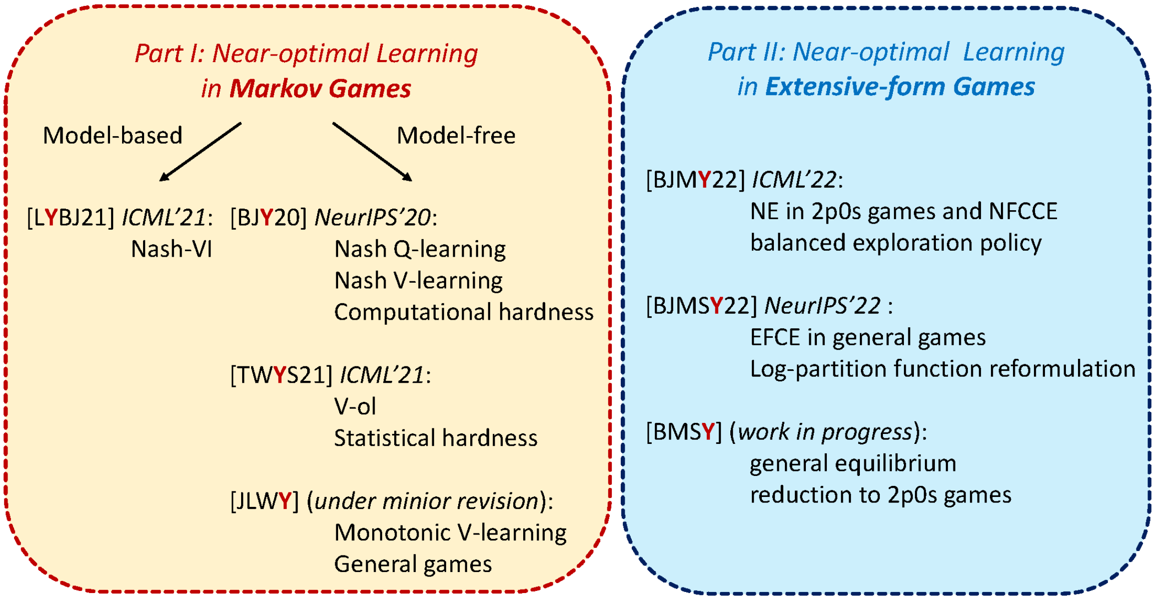

Part I: Near-optimal Learning in Markov Games
Model-based
Model-free
[LYBJ21] ICML'21: Nash-VI
[BJY20] NeurIPS'20: Nash Q-learning Nash V-learning Computational hardness
[TWYS21] ICML'21: V-ol Statistical hardness
[JLWY] (under minior revision): Monotonic V-learning General games
Part II: Near-optimal Learning in Extensive-form Games
[BJMY22] ICML'22: NE in 2p0s games and NFCCE balanced exploration policy
[BJMSY22] NeurIPS'22 : EFCE in general games Log-partition function reformulation
[BMSY] (work in progress): general equilibrium reduction to 2p0s games

Figure 1-5: A roadmap of the results presented in this section.

learning in two of the best-known examples of sequential games: Markov games (MGs) and Extensive-form games (EFGs).

Remark 1. *From now on, we will use the wording "player" instead of "agent" and "return" instead of "reward". These choice is more consistent with the convention of game theory, but essentially makes no difference in mearning.*

1.1 Overview of results

Before delving into the detailed problem formulations and solutions, we first go through the main results of this thesis, and compare with the existing results. This section serves as both a result overview and a guidance on the content of each chapters. See Figure 1-5 for a graphical roadmap.

All of the content in this thesis has already been published in peer-review conferences, except Jin et al. [2021b] which is under minor review by a jorunal at the time when this thesis is written. We give pointers to the original papers when the corresponding result is presented below.

1.1.1 Near-optimal learning in Markov games

In Chapter 2 we introduce the learning problem formulation for Markov games. We present two hardness results: one computational lower bound from Bai et al. [2020] and one statistical lower bound from Tian et al. [2021]. This rules out the possibility of controlling the sample complexity through regret minimization directly, which is the common practice in the literature of learning in games.

To break the regret minimization barrier, we propose two different workarounds.

- In Chapter 3, we coordinate all the players through a centralized model estimation to minimize the regret. By carefully quantifing the estimation uncertainty in a novel way and combine the estimated model with the calssical value iteration, we propose a new algorithm called *Nash value iteration* (Nash-VI). Because we explicitly estiamte the transition (model), the approach is referred to as *model-based* in literature. The content of this chapter is based on Liu et al. [2021].

- In Chapter 4, we introduce a new technique called *certified policy* to combine historical policies in a way different from the conventional online-to-batch conversion. Using the policies generated by local regret minimization subprocedures, we propose a new algorithm called Nash V-learning. Since we do not estimate the transition explicitly but only through updating certain value function, the approach is referred to as *model-free*. The content of this chapter is based on Bai et al. [2020], Tian et al. [2021], Jin et al. [2021b].

The new methods (colored in light gray) are compared with existing results in Table 1.1. Here we consider the sample complexity of learning an ε-approximation of a Nash equilibirum in a two-player zero-sum Markov games with S states, A actions for the max-player, B actions for the min-players and horizon H. See a formal definition of Nash equilibira in Section 2.

The main difference between the model-based and model-free methods is: the model-based method achieves better dependence on the horizon while the model-free method improves $\tilde{\mathcal{O}}(AB)$ to $\tilde{\mathcal{O}}(A+B)$. To see why this improvement is significant, we

Method	Sample complexity
VI-ULCB [Bai and Jin, 2020]	$\tilde{\mathcal{O}}\left(H^4S^2AB/\varepsilon^2\right)$
OMVI-SM [Xie et al., 2020]	$\tilde{\mathcal{O}}\left(H^4S^3A^3B^3/\varepsilon^2\right)$
Nash Q-learning [Bai et al., 2020]	$\tilde{\mathcal{O}}\left(H^5SAB/\varepsilon^2\right)$
Nash-VI [Liu et al., 2021]	$\tilde{\mathcal{O}}\left(H^3SAB/\varepsilon^2\right)$
Nash V-learning [Bai et al., 2020, Jin et al., 2021b]	$\tilde{\mathcal{O}}\left(H^5S(A+B)/\varepsilon^2\right)$
Lower bound [Bai and Jin, 2020]	$\Omega\left(H^3S(A+B)/\varepsilon^2\right)$

Table 1.1: Comparison with existing results for two-player zero-sum Markov games. The Nash-VI algorithm matches the lower bound in terms of the dependence on horizon while the Nash V-learning algorithm matches the lower bound w.r.t. the other key parameters.

Method	Correlated Equilibrium (CE)	Coarse Correlated Equilibrium (CCE)
Nash-VI [Liu et al., 2021]	$\tilde{\mathcal{O}}\left(H^4S^2\prod_i A_i/\varepsilon^2\right)$	$\tilde{\mathcal{O}}\left(H^4S^2\prod_i A_i/\varepsilon^2\right)$
Nash V-learnig [Jin et al., 2021b]	$\tilde{\mathcal{O}}\left(H^5S\max_i A_i/\varepsilon^2\right)$	$\tilde{\mathcal{O}}\left(H^5S\max_i A_i^2/\varepsilon^2\right)$

Table 1.2: Comparison of model-based and model-free method in multi-player general-sum Markov games.

need to think about the more general multi-player games, where each player $i \in [m]$ has an action set of size A_i. We present the sample complexity of learning correlated equilibrium and coarse correlated equilibrium in Table 1.2.

The sample complexity of model-based method grows exponentially as the number of players m increases, while the sample complexity of model-free method only depends on the maximum of the action set across players, which is much more favorable in the presence of a large number of players.

To conclude the results of learning in MGs, let us make a more detailed comparison between Nash-VI and Nash V-learning (See Table 1.2). In Nash VI, there is a centralized estimator and planner, to first estimate the MG based on observations from both players, and then compute the equilibria and inform all the players. On the contrary, Nash V-learning is decentralized, and each player updates her own decision rule based on her observations. Since the underlying MG is not estimated explicitly, V-learning is a model-free algorithm. See Figure 1-6 for an intuitive comparison

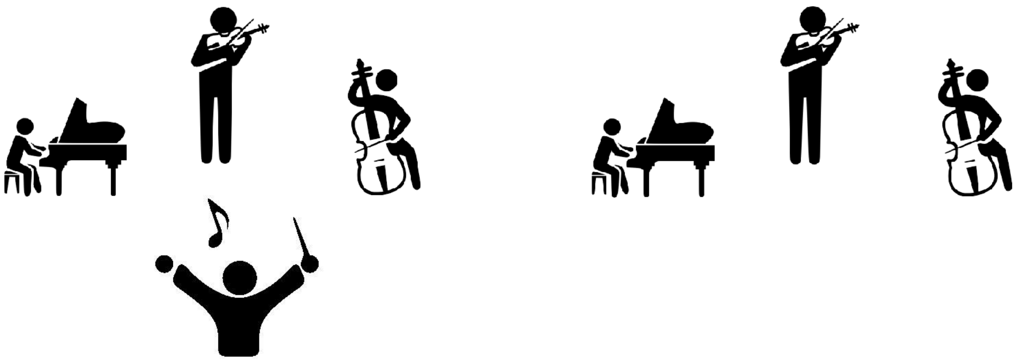


Figure 1-6: An intuitive comparison between Nash-VI (left) and Nash V-learning (right): The Nash-VI algorithm is more like an orchestra. There is a conductor – the centralized estimator and planner, to first estimate the MG based on observations from both players, and then compute the NE and inform all the players. On the contrary, Nash V-learning is more like chamber music where each player update her own decision rule based on her own observations.

between model-based and model-free methods.

As we have seen from the sample complexity result, Nash-VI is preferable for longer horizons, while V-learning works better given more players or larger action sets. Also, since centralized model estimation and equilibrium computation is not needed, V-learning embraces a simpler and symmetric update rule for each player, which makes the deployment easier, communication-free, and computationally more efficient.

1.1.2 Near-optimal learning in extensive-form games

In Chapter 5 we introduce the learning problem formulation for imperfect-information extensive-form games. A lower bound from Bai et al. [2022b] is presented to characterized the fundamental limit of learning an ε-Nash equilibrium in extensive-form games.

In Chapter 6, we study the near-optimal learning algorithms under the two-player zero-sum game setting. The content is based on Bai et al. [2022b] and the resulting sample compelxity is compared with the exsiting methods in the Table 1.3. Here we consider learning an ε-Nash Equilibrium in a two-player zero-sum Extensive-form game, with X information sets and A actions for the max player, Y information sets

Algorithm	**OMD**	**CFR**	**Sample Complexity**
Zhang and Sandholm [2021]	- (model-based)		$\widetilde{\mathcal{O}}\left(S^2AB/\varepsilon^2\right)$
Farina and Sandholm [2021]		✓	$\widetilde{\mathcal{O}}(\text{poly}\left(X, Y, A, B\right)/\varepsilon^4)$
Farina et al. [2021b]	✓		$\widetilde{\mathcal{O}}\left(\left(X^4A^3 + Y^4B^3\right)/\varepsilon^2\right)$
Kozuno et al. [2021]	✓		$\widetilde{\mathcal{O}}\left(\left(X^2A + Y^2B\right)/\varepsilon^2\right)$
Balanced OMD (Algorithm 10)	✓		$\widetilde{\mathcal{O}}\left(\left(XA + YB\right)/\varepsilon^2\right)$
Balanced CFR (Algorithm 11)		✓	$\widetilde{\mathcal{O}}\left(\left(XA + YB\right)/\varepsilon^2\right)$
Lower bound (Theorem 35)	-	-	$\Omega\left(\left(XA + YB\right)/\varepsilon^2\right)$

Table 1.3: Sample complexity (number of episodes required) for learning ε-NE intwo-player zero-sum Extensive-form games from bandit feedback.

and B actions for the min player, and S unobservable hidden states.

Most of the algorithms in the table are either an instance of online mirror descent (OMD) or counterfectual regret minimization (CFR). See Chapter 6 for a more detailed description on these two types of algorithms. The only exception is the first row, which is a model-based algorithm. The disadvantage is that it requires some "cheating" power in the training process by observing the state directly, and its sample complexity depends on S instead of X and Y. It could be the case that X and Y are very small but S is very large, which makes the form of sample complexity not favorable.

The two rows in gray are algorithms proposed in our work. They combine a novel technique called **balanced exploration policy** with OMD and CFR separately. Comparing the upper and lower bounds, both balanced OMD and balanced CFR could achieve near-optimal sample complexity. They are the first line of algorithms that achieve linear dependence on the number of information sets, which answers an open question proposed in Kozuno et al. [2021].

In Chapter 7 we exntend the above sharp sample complexity bounds to multi-player general-sum games. We develop the first line of learning Extensive-form correlated equilibria (EFCEs) with near-optimal sample complexity. The result is a consequence of dual reformulation, by interpreting the calssical Φ-hedge algorithm

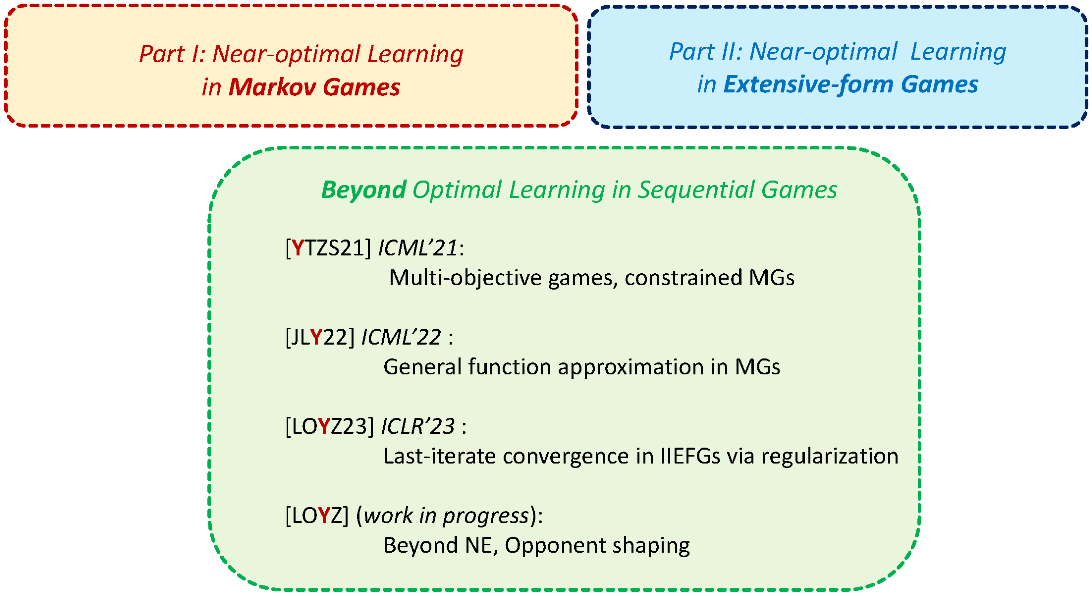

Part I: Near-optimal Learning in Markov Games
Part II: Near-optimal Learning in Extensive-form Games
Beyond Optimal Learning in Sequential Games
[YTZS21] ICML'21: Multi-objective games, constrained MGs
[JLY22] ICML'22 : General function approximation in MGs
[LOYZ23] ICLR'23 : Last-iterate convergence in IIEFGs via regularization
[LOYZ] (work in progress): Beyond NE, Opponent shaping

Figure 1-7: A raod map of the results presented in this section.

as computing gradient of a log-partition function. The reformulation helps extend algorithms on normal-form games to EFGs. Besides improving the sample efficiency, it also simplifies the presentation and proof of some existing results and helps us discover new connections between known algorithms. The content of this chapter is based on Bai et al. [2022a].

1.1.3 Beyond optimal learning in sequential games

Although sample efficiency is arguably the most fundamental problem for RL, it is by no means the only important problem. There are some other exciting projects that I pursued during my PhD years, on other aspects of learning in sequential games. To make the presentation of this thesis more succinct, I decided not to include them in full detail, but to give a brief overview here for those who might be interested. See Figure 1-7 for a graphical roadmap.

Mutli-objective games. So far we only consider a scalar utility function and unconstrained problem. Sometimes we have multiple (vectorized) utility function or a constraint to satisfy. See Figure 1-8 for an illustration of the learning objective.

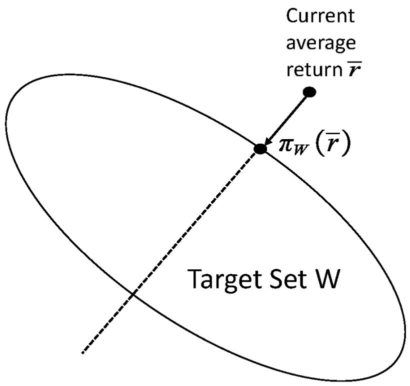


Figure 1-8: An illustration of the Blackwell approachability approach.

Figure 1-9: Tabular RL (left) v.s. RL with function approximation (right).

This is considered in Yu et al. [2021], by extending the classical notion of Blackwell approachability [Blackwell, 1956] to the RL setting. The framework can also be applied to constrained MGs to develop near-optimal sample complexity.

Function approximation. In either self-play or online learning setting, if the number of states is vast, then even maintaining a policy for each state is impossible. Therefore one has to use function approximation. See Figure 1-9 for a comparison between the tabular and function approximation settings for RL.

We first distinguish a key complexity measure, multi-agent Bellman-Eluder dimension for Markov games Jin et al. [2022]. Using multi-agent Bellman-Eluder dimension to quantify the complexity of the function class, we design a novel exploiter-based algorithm for learning the Nash equilibrium with sharp sample complexity. In each episode, the exploiter serves as the "strongest" opponent to facilitate the learning of our agent by pessimistic planning based on value function elimination.

Last iterate guarantee. All of the algorithms considered in this thesis have the following feature: we take different policies in different episodes and average them in a certain way to find an output policy that is guaranteed to be an approximate Nash equilibrium policy. A more ambitious goal is to make this guarantee on the policy used in the last episode directly, without any averaging scheme. This motivates a long line of work [Wei et al., 2021a,b, Lee et al., 2021, Daskalakis and Panageas, 2019, Golowich et al., 2020b,a, Cai et al., 2022, Cen et al., 2023]. In particular our work Liu et al. [2023] takes a regularization perspective and achieve this goal in IIEFGs.

1.2 Related work

In this section, we overview the literature on learning in sequential games.

1.2.1 Learning in Markov games

Although there has been much recent work in RL for continuous state spaces [see, e.g., Jiang et al., 2017, Jin et al., 2020c, Zanette et al., 2020, Jin et al., 2021a, Xie et al., 2020, Jin et al., 2022], this setting is beyond our scope. We will focus on theoretical results for the tabular setting, where the numbers of states and actions are finite.

Markov games. Markov Game (MG), also known as stochastic game [Shapley, 1953], is a popular model in multi-agent RL [Littman, 1994]. Early works have mainly focused on finding Nash equilibria of MGs under strong assumptions, such as known transition and reward [Littman, 2001, Hu and Wellman, 2003, Hansen et al., 2013, Wei et al., 2021a], or certain reachability conditions [Wei et al., 2017, 2021b] (e.g., having access to simulators [Jia et al., 2019, Sidford et al., 2020, Zhang et al., 2020a]) that alleviate the challenge in exploration. We remark that, while Wei et al.

[2017] studies the infinite-horizon average-reward setting, a recently refined analysis tailored to the episodic setting shows that the algorithms proposed in Wei et al. [2017] can learn Nash equilibria in two-player zero-sum MGs without additional reachability assumptions in $\tilde{\mathcal{O}}(H^4 S^2 A_1 A_2/\varepsilon^2)$ [Wei, 2021].

A recent line of works provides non-asymptotic guarantees for learning two-player zero-sum tabular MGs without further structural assumptions. Bai and Jin [2020] and Xie et al. [2020] develop the first provably-efficient learning algorithms in MGs based on optimistic value iteration. Liu et al. [2021] improves upon these works and achieves best-known sample complexity for finding an ε-Nash equilibrium—$\mathcal{O}(H^3 S A_1 A_2/\varepsilon^2)$ episodes.

For multiplayer general-sum tabular MGs, Liu et al. [2021] is the only existing work that provides non-asymptotic guarantees in the exploration setting. It proposes centralized model-based algorithms based on value iteration, and shows that Nash equilibria (although computationally inefficient), CCE and CE can be all learned within $\mathcal{O}(H^4 S^2 \prod_{j=1}^{m} A_j/\varepsilon^2)$ episodes. Note this result suffers from the curse of multiagents.

V-learning, initially coupled with the FTRL algorithm as adversarial bandit subroutine, was first proposed in Bai et al. [2020], for finding Nash equilibria in the two-player zero-sum setting. During the preparation of the submission Jin et al. [2021b], we noted two very recent independent works Song et al. [2022a], Mao and Başar [2023], whose results partially overlap with the results of Jin et al. [2021b] in the multiplayer general-sum setting. In particular, Mao and Başar [2023] use V-learning with stabilized online mirror descent as adversarial bandit subroutine, and learn ε-CCE in $\mathcal{O}(H^6 S A/\varepsilon^2)$ episodes, where $A = \max_{j\in[m]} A_j$. The complexity is H times larger than what is required in Theorem 24. Song et al. [2022a] considers similar V-learning style algorithms for learning both ε-CCE and ε-CE. For the latter objective, they require $\mathcal{O}(H^6 S A^2/\varepsilon^2)$ episodes which is again one H factor larger than what is required in Theorem 25. We remark that both parallel works have not presented V-learning as a generic class of algorithms which can be coupled with any adversarial bandit algorithm with suitable regret guarantees in a black-box fashion.

We also notice a number of related works that appeared after the initial conference publication of Bai et al. [2020], including [Kao et al., 2022, Mao et al., 2022, Liu et al., 2022, Zhan et al., 2023, Erez et al., 2022, Daskalakis et al., 2022]. Kao et al. [2022] consider fully cooperative two-player games with a sequential structure with the goal of learning the joint optimal policy in a decentralized fashion. This is a drastically different setting and is not directly comparable to this work. Mao et al. [2022] propose a stage-based variant of V-learning algorithm so that any standard no-regret adversarial bandit algorithm can be used in place of the no-weighted-regret algorithm employed in Jin et al. [2021b]. Their modified algorithms recover the rates in Theorem 24 and 25. Liu et al. [2022] and Zhan et al. [2023] consider the complexity of no-regret learning against adversarial opponents for tabular and function approximation Markov games respectively. Their positive result mostly applies when competing against the best Markov policy in hindsight, while in Jin et al. [2021b] our definition of equilibria compares with the best general policy in hindsight. Erez et al. [2022] develop a computationally efficient algorithm for learning CE, but requires additional structural assumptions to prove statistical efficiency. Daskalakis et al. [2022] consider a more challenging task than the one addressed in Jin et al. [2021b], namely learning *Markov CCE* — CCE policies that are history-independent given the current state. As a trade-off for their stronger solution concept, their algorithm is much more sophisticated and requires a larger number of samples.

Normal-form games. A normal-form game (NFG) is one of the most basic game forms studied in the game theory literature Osborne and Rubinstein [1994]. It can be viewed as Markov games *without* any states or transition. A fully decentralized algorithm that breaks the curse of multiagents is known in the setting of strategic-form games. By independently running a no-regret (or no-swap-regret) algorithm for all agents, one can find Nash Equilibria (in the two-player zero-sum setting), correlated equilibria and coarse correlated equilibria (in the multiplayer general-sum setting) in a number of samples that only scales with $\max_{i\in[m]} A_i$ Cesa-Bianchi and Lugosi [2006], Hart and Mas-Colell [2000], Blum and Mansour [2007]. However, such successes do not directly extend to the Markov games due to the additional temporal structures involving both states and transition. In particular, there is no computationally efficient no-regret algorithm for Markov games Radanovic et al. [2019], Bai et al. [2020].

Decentralized MARL. There is a long line of *empirical* works on decentralized MARL [see, e.g., Lowe et al., 2017, Iqbal and Sha, 2019, Sunehag et al., 2018, Rashid et al., 2018, Son et al., 2019]. A majority of these works focus on the cooperative setting. They additionally attack the challenge where each agent can only observe a part of the underlying state, which is beyond the scope of Jin et al. [2021b]. For theoretical results, Zhang et al. [2018] consider the cooperative setting while Sayin et al. [2021] study the two-player zero-sum Markov games. Both develop decentralized MARL algorithms but provide only asymptotic guarantees. Daskalakis et al. [2020] analyze the convergence rate of independent policy gradient method in episodic two-player zero-sum MGs. Their result requires the additional reachability assumptions (concentrability) which alleviates the difficulty of exploration.

Single-agent RL. There is a rich literature on reinforcement learning in MDPs [see e.g., Jaksch et al., 2010, Osband et al., 2016, Azar et al., 2017, Dann et al., 2017, Strehl et al., 2006, Jin et al., 2018, 2021a]. MDPs are special cases of Markov games, where only a single agent interacts with a stochastic environment. For the tabular episodic setting with nonstationary dynamics and no simulators, the best sample complexity achieved by existing model-based and model-free algorithms are $\tilde{\mathcal{O}}(H^3SA/\varepsilon^2)$ (achieved by value iteration Azar et al. [2017]) and $\tilde{\mathcal{O}}(H^4SA/\varepsilon^2)$ (achieved by Q-learning Jin et al. [2018]), respectively, where S is the number of states, A is the number of actions, H is the length of each episode. Both of them (nearly) match the lower bound $\Omega(H^3SA/\varepsilon^2)$ Jaksch et al. [2010], Osband and Van Roy [2016], Jin et al. [2018].

1.2.2 Learning in extensive-form games

Regret minimization in EFG from full feedback. A line of work considers external regret minimization in EFGs from full feedback [Zinkevich et al., 2007, Celli et al., 2019b, Burch et al., 2019, Farina et al., 2021a, Zhou et al., 2020]. In particular, Zhou et al. [2020] achieves $\widetilde{\mathcal{O}}(\sqrt{XT})$ external regret. The recent work of Farina et al. [2022b] develops the first algorithm to achieve $\widetilde{\mathcal{O}}(\|\Pi\|_1 \text{polylog} T)$ external regret in EFGs by converting it to an NFG and invoking the fast rate of Optimistic Hedge [Daskalakis et al., 2021], along with an efficient implementation via the "kernel trick". Our Φ-regret framework covers their algorithm as a special case, and we further show that their algorithm (along with its efficient implementation) is equivalent to the standard OMD with dilated entropy.

The notion of Extensive-Form Correlated Equilibria (EFCE) in EFGs was introduced in Von Stengel and Forges [2008]. Optimization-based algorithms for computing computing EFCEs in multi-player EFGs from full feedback have been proposed in Huang and von Stengel [2008], Farina et al. [2019].

Gordon et al. [2008] first proposed to use uncoupled EFCE-regret minimization dynamics to compute EFCE; however, they do not explain how to efficiently implement each iteration of the dynamics. Recent works Celli et al. [2020], Farina et al. [2022a], Morrill et al. [2021], Song et al. [2022b] developed uncoupled EFCE regret minimization learning dynamics with efficient implementation; All of these algorithms are based on counterfactual regret decomposition [Zinkevich et al., 2007] and minimizing each trigger regret (first considered by Dudík and Gordon [2009], Gordon et al. [2008]) using a different regret minimizer. Celli et al. [2020] decomposed the regret to each laminar subtree, but they did not give an explicit regret bound. Farina et al. [2022a] decomposed the regret to each trigger sequence and used CFR type algorithm to minimize the regret on each trigger sequence and achieved an $\tilde{O}(\sqrt{X^2T})$ EFCE-regret bound. Morrill et al. [2021], Song et al. [2022b] decomposed the regret to each information set and use regret minimization algorithms with time-selection functions Blum and Mansour [2007], Khot and Ponnuswami [2008] to minimize the regret on each information set, giving $\widetilde{\mathcal{O}}(\sqrt{X^2T})$ and $\widetilde{\mathcal{O}}(\sqrt{XT})$ regret bounds respectively. In Bai et al. [2022a], we show that the simple Φ-Hedge algorithm, which has an efficient implementation and an intuitive interpretation, can also achieve the state-of-art $\widetilde{\mathcal{O}}(\sqrt{XT})$ regret bound in the full feedback setting.

Regret minimization in EFG from bandit feedback. Minimizing the external regret in EFGs from bandit feedback is considered in a more recent line of work [Lanctot et al., 2009, Farina et al., 2020b, Farina and Sandholm, 2021, Farina et al., 2021b, Zhou et al., 2019, Zhang and Sandholm, 2021, Kozuno et al., 2021, Bai et al., 2022b]. Dudík and Gordon [2009] consider sample-based learning of EFCE in succinct extensive-form games; however, their algorithm relies on an approximate Markov-Chain Monte-Carlo sampling subroutine that does not lead to a sample complexity guarantee.

A concurrent work by Song et al. [2022b] also achieves $\widetilde{\mathcal{O}}(X/\varepsilon^2)$ sample complexity for learning EFCE under bandit feedback (when only highlighting X) using the Balanced K-EFR algorithm. Our work achieves the same linear in X sample complexity, but using a very different algorithm (Balanced EFCE-OMD). We also remark that the algorithm of [Song et al., 2022b] cannot minimize the EFCE-regret against adversarial opponents from bandit feedback like our algorithm, as their algorithm requires playing multiple episodes against a fixed opponent, which is infeasible when the opponent is adversarial.

Φ-regret minimization and correlated equilibrium. The Φ-regret minimization framework was introduced in Greenwald and Jafari [2003] and Stoltz and Lugosi [2007]. In particular, Greenwald and Jafari [2003] showed that uncoupled no Φ-regret dynamics leads to Φ-correlated equilibria, a generalized notion of correlated equilibria introduced by Aumann [1974]. Stoltz and Lugosi [2007] then developed a family of Φ-regret minimization algorithms using the fixed-point method (including the Φ-Hedge algorithm considered in Bai et al. [2022a]), and derived explicit regret bounds. Two important special cases of Φ-regret are the internal regret and swap regret in normal-form games Stoltz and Lugosi [2005], Blum and Mansour [2007]. A recent line of work developed algorithms with $\mathcal{O}(\text{polylog}T)$ swap regret bound in normal-form games Anagnostides et al. [2022a,b].

1.3 Open problems and future work

Despite the endeavour of the community of learning in sequential games and the progress in recent years, there are still many issue remaining unclear. In this section, I will overview some of the most relevant and exiciting open problems in learning in sequential games.

1.3.1 Remaining issues in optimal learning in sequential games

We begin with some technical issues that are most relevant to the scope of the thesis. Interestingly, most of the gap is related to one common theme: the model-based approach achieves better dependence on T (or ε) and H, and model-free approach achieves better dependence on the number of actions of each players. Is there a way to achieve the best-of-both-worlds?

Optimal sample complexity of learning in MGs. As presented in Table 1.1, Nash-VILiu et al. [2021] achieves optimal dependence on S and H, but not on A and B, while Nash V-learning achieves optiaml dependence on S, A and B, but not on H. Is it possible to improve either of the two algorithms to achieve optimality for all of the parameters?

For model-based methods, intuitively since the size of the model depends on AB, it seems very unlikely that we can actually achieve $A+B$ sample complexity. However it is also hard to rule out this possibility rigorously because it is not clear how to define the class of model-based algorithm formally to prove a lower bound. For example, if one just runs V-learning but estimates the model at the same time, does it count as "model-based"? We indeed have some negative result, though. In Liu et al. [2021], we showed that the sample complexity of a closely related problem, reward-free learning, is at least AB.

For model-free methods, the current gap seems more like an artifact. In the single-agent setting, there are some techniques to improve the dependence on horizon for model-free algorithms. For example, using a Bernstein-type optimistic bonus Jin et al. [2018] and a reference estimation [Zhang et al., 2020c] can both shave off an H factor in the sample complexity bound. However, in the game setting, it is not at all clear how to combine these techniques with regret minimization procedure as we have used in V-learning type of algorithms.

Optimal learning in potential MGs. There is one important instance of multi-player general-sum games called **potential games** [Monderer and Shapley, 1996] that characterizes players with a shared utility, and whose Nash equilibrium can be found in polynomial time. A recent line of work focuses on sample-efficient learning in Markov potential game and in particular Song et al. [2022a] develops a $\mathcal{O}(\varepsilon^{-3})$ sample complexity bound, and the dependence on the other paramters is polynomial, which is favorable when the number of players increases. In each episode, the algorithms re-collect data to estimate the value function, and much of the data are not reused later, which results in the sub-optimal $\mathcal{O}(\varepsilon^{-3})$ bound. On the other hand, we can use the general algorithm as in Liu et al. [2021] to achieve $\mathcal{O}(\varepsilon^{-2})$ sample complexity, but the dependence on the other paramters is exponential, as a consequence of the model-based planning.

Naturally, one may wonder if we can achieve best of both world: achieving $\mathcal{O}(\varepsilon^{-2})$ sample complexity and polynomial dependence on the other parameters at the same time. Unluckily the answer is unclear at the time of this thesis.

A closely realted setting is congestion game [Milchtaich, 1996], where the question can be answered affirmatively in Cui et al. [2022]. However, the classical reduction from a congestion game to a potential game may involve an exponential blow-up, so this positive result itself cannot answer the above question directly.

Online learning in MGs. Here comes the third example of the best-of-both-worlds type of question, but in a online learning setting. If we consider online learning in Markov games, as we will show in Section 2.5, competing with optimal policy in hindsight is provably hard. Hence we can only expect to compete with the Nash equilibirum policy, which serves as a stationary and thus weaker baseline.

However, even in this scenario, the minimax regret is still open. Following a model-based approach as in Liu et al. [2021], we can achieve $\mathcal{O}\sqrt{T}$ regret but the dependence on the number of players is exponential. On the other hand, following the V-OL algorithm in Section 4.5.1, the sample complexity is almost free of the number players involved, but the dependence on T is only $\mathcal{O}(T^{2/3})$. Again, the open question is: can we achieve $\mathcal{O}\sqrt{T}$ regret and polynomial dependence on the number of players?

Fully model-free learning. The last open problem regrading optimal sample complexity has a different flavor. In the original formulation of the balanced exploration policy as introduced in Section 6.1, we have to know the game tree stucture in advance (at least from each players' perspective), to design such policies. In a follow-up paper Fiegel et al. [2022], a surrogate is proposed to avoid this issue. In particular, the surrogate only depends on information sets that has been visited so far and the historical policies. However, the cost is to pay an additional H factor in sample complexity. This induces an interesting question, whether knowing the tree structure changing the difficulty of the problem. If so, how much is the gap and how should we understand the value of knowing the tree structure?

1.3.2 Beyond tabular sequential games

All of the results presented in this thesis are focusing on the tabular setting, where the number of state and actions are finite. However in many large-scale problems, even though the number of state and actions are finite, it is formidably large so that even linear dependence on the number of state and actions is unrealistic. The standard approach to handle such large state spaces is by using function approximation. As long as the size of the function class (usually in terms of a certain kind of covering number) is controllable and the sample complexity has a sharp dependence (hopefully linear) on the size, then function approximation approach could give a much more practical and scalable solution. In this section, we discuss the challenge of extending the results in tabular setting to the function approximation setting.

Statistical-computational gap of general function approximation. Although Jin et al. [2022] achieving near-optimal satistical efficiency in many problems, the implementation involves solving a complex optimization problem. This problem appears even in the single agent setting. In fact, none of the existing general function approximation RL algorithms with polynomial sample complexity guarantee, such as Jin et al. [2021a], Du et al. [2021], Foster et al. [2021] can be implemented in polynomial time. Hence a natural question is: does the statistical-computational gap exist for RL with function approximation?

There are two standard assumptions used in the analysis of the function approximation RL algorithms:

- **Realizability**: the ground truth value function or Q-value function can indeed be approximated the function class.

- **Completeness**: the function class itself has some closeness property such that the value function from the class will also induce some other value function from the class.

Only under the realizability condition, there is a line of research demonstrating the computational hardness of the problem when the function class only contains linear function. In particular, as pointed out in Kane et al. [2022], the statistical-computational gap exists. However, when completeness is also enforced, it is not clear if such a gap still exists.

On the other hand, one may wonder if the hard optimization problem introduce in Jin et al. [2022] can be solved either in certain special cases or heuristically. This is an exciting question that I have devoted a lot of effort to during my PhD time but did not make much progress. In practice one prefers to try out simiplified exploration scheme.

Linearly parametrized sequential games. There is indeed a well-studied function approximation setting for RL algorithms, at least for single agent scenario —- linear MDP (or low-rank MDP) Jin et al. [2020c]. We already know only linearly parametrized value function is not enough. In a linear MG, the whole transition and reward are fully linearly parametrized and the completeness condition is naturally satisfied. Even better, the optimization problem can be solved efficiently so Jin et al. [2022] translates to an computationally efficient algorithms, which is not that surprising because Xie et al. [2020] already contains a similar result for linear MG with a worse sample complexity.

A more interesting and also ambitious task is to extend the result in Bai et al. [2020], i.e., the Nash Q-learning and Nash V-learning algorithms, to the linear MG setting, which is partially solved by a recent work by Wang et al. [2023]. One important difference in their assumptions is to use “marginalized” feature and make the transition and reward multi-linearly depending on the features.

Principled function approximation in EFGs. When the state itself is not obsevable, we do not have a optimal substructure property, which makes principled function approximation much harder. In MGs we can approximate the optimal value function for each state action pair, but such quantity even does not exist for EFGs. Maybe the most fundamental question is: what do we really want to approximate here? Practitioners approximate the counterfactual values through neural networks, but these quantities depend on the policy used and keep changing all the time. As a result, it is hard to give any theoretical guarantee for such function approximation algorithms.

1.3.3 Beyond Nash equilibrium: learning to cooperate

All of the techniques we present in the thesis serves the same purpose: learning not to regret, which can be translated to certain equilibrium guarantees. However, finding an equilibrium is not the whole story of learning in games. In many case we also want to learn to cooperate.

The most famous example is the prisoner's dilemma. It is well known that the only Nash equilibrium is to betray all the time. However, if both players can learn to cooperate, then the average utility will be much higher. Of course, this is only possible if the other player is willing to cooperate. The question is: how can we adjust to different type of player, embrace cooperation when possible and avoid being exploited by "bad" players?

This is a fascinating problem that I have given a lot of thoughts to during the past years. To solve this problem we have to first introduce a new framework and make it a well-defined question. Conventional notions of regret, Nash equilibrium and social welfare may still be relevant, but we definitely need to introduce some new measures of the adaptivity we want.

At the end of the day, the ideal algorithm should be able to satisfy two conditions: it is able to adapt to different type of players and when it is deployed on both players, it should facilitate the players to cooperate efficiently. I believe such "learning to cooperate" power is a crucial component of the general artificial intelligence sought by researchers.

Chapter 2

Markov Games: Preliminaries

Modern artificial intelligence (AI) faces a variety of challenges that can be framed as multi-agent reinforcement learning (MARL) problems. In MARL, agents must learn how to make a series of decisions in the presence of other agents whose actions may impact the outcome. This dynamic environment requires agents to adapt their strategies to those of other agents, putting an additional layer of complexity to the problem.

Modern MARL systems have achieved significant success recently on a rich set of traditionally challenging tasks, including the game of GO [Silver et al., 2016, 2017], Poker [Brown and Sandholm, 2019], real-time strategy games [Vinyals et al., 2019, OpenAI, 2018], decentralized controls or multiagent robotics systems [Brambilla et al., 2013], autonomous driving [Shalev-Shwartz et al., 2016], as well as complex social scenarios such as hide-and-seek [Baker et al., 2020]. While single-agent RL has been the focus of recent intense theoretical study, MARL has been comparatively underexplored, which leaves several fundamental questions open even in the basic model of *Markov games* Shapley [1953] with finitely many states and actions.

This chapter presents an overview of Markov games (MGs), including their formulation and associated learning problems. The framework is accompanied by two hardness results, ruling out the possibility of achiving no-regret learning in Markov games in general.

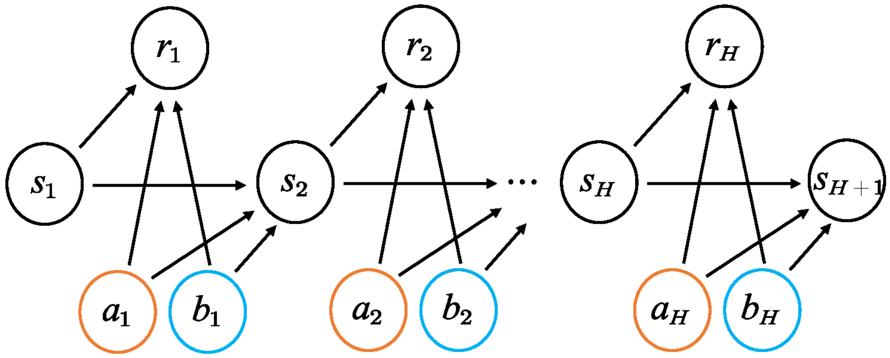


Figure 2-1: An illustration of MGs.

2.1 Game formulation

We consider the model of Markov Games [Shapley, 1953] (also known as stochastic games in the literature) in its most generic—multiplayer general-sum form. Formally, we denote a tabular episodic MG with m players by a tuple $\mathrm{MG}(H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathbb{P}, \{r_i\}_{i=1}^m)$, where H and $\mathcal{S}$ denote the length of each episode and the state space with $|\mathcal{S}| = S$. $\mathcal{A}_i$ denotes the action space for the $it, (h)$ player and $|\mathcal{A}_i| = A_i$. We let $\boldsymbol{a} := (a_1, \cdots, a_m)$ denote the (tuple of) joint actions by all m players, and $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_m$. $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is a collection of transition matrices, so that $\mathbb{P}_h(\cdot|s, \boldsymbol{a})$ gives the distribution of the next state if actions $\boldsymbol{a}$ are taken at state s at step h, and $r_i = \{r_{i,h}\}_{h\in[H]}$ is a collection of reward functions for the i^{th} player, so that $r_{i,h}(s, \boldsymbol{a}) \in [0, 1]$ gives the deterministic reward received by the i^{th} player if actions $\boldsymbol{a}$ are taken at state s at step h. Our results directly generalize to random reward functions, since learning transitions is more difficult than learning rewards. See Figure 2-1 for an illustration of the decision-making process descibed.

We remark that since the relation among the rewards of different agents can be arbitrary, this model of MGs incorporates both cooperation and competition.

In each episode, we start with a *fixed initial state* s_1. While we assume a fixed initial state for notational simplicity, our results readily extend to the setting where the initial state is sampled from a fixed initial distribution. At each step $h \in [H]$, each player i observes state $s_h \in \mathcal{S}$, picks action $a_{i,h} \in \mathcal{A}_i$ simultaneously, and receives her own reward $r_{i,h}(s_h, \boldsymbol{a}_h)$. Then the environment transitions to the next state $s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, \boldsymbol{a}_h)$. The episode ends when s_{H+1} is reached.

Notice each agent may or may not observe the actions played by other players in this process. When the actions of the other players are observed, we call it an **informed** game [Tian et al., 2021], and otherwise we call it an **unknown** game [Cesa-Bianchi and Lugosi, 2006]. The model-based approach introduced in Chapter 3 only works in informed games while the model-free approach introduced in Chapter 4 also works in unknown games.

2.1.1 Policy

A (*random*) *policy* π_i of the i^{th} player is a set of H maps $\pi_i := \big\{\pi_{i,h} : \Omega \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \mathcal{A}_i\big\}_{h\in[H]}$, where $\pi_{i,h}$ maps a random sample ω from a probability space Ω and a history of length h—say $\tau_h := (s_1, \boldsymbol{a}_1, \cdots, s_h)$, to an action in $\mathcal{A}_i$. To execute policy π_i, we first draw a random sample ω at the beginning of the episode. Then, at each step h, the $it,(h)$ player simply takes action $\pi_{i,h}(\omega, \tau_h)$. We note here ω is shared among all steps $h \in [H]$. ω encodes both the correlation among steps and the individual randomness of each step. We further say a policy π_i is *deterministic* if $\pi_{i,h}(\omega, \tau_h) = \pi_{i,h}(\tau_h)$ which is independent of the choice of ω.

In game theory literature, policies are usually called strategies. To be consitent with some common terms, we will use these two words interchangably when needed.

An important subclass of policy is *Markov policy*, which can be defined as $\pi_i := \big\{\pi_{i,h} : \Omega \times \mathcal{S} \to \mathcal{A}_i\big\}_{h\in[H]}$. Instead of depending on the entire history, a Markov policy takes actions only based on the current state. Furthermore, the randomness in each step of Markov policy is independent. Therefore, when it is clear from the context, we write Markov policy as $\pi_i := \big\{\pi_{i,h} : \mathcal{S} \to \Delta_{\mathcal{A}_i}\big\}_{h\in[H]}$, where $\Delta_{\mathcal{A}_i}$ denotes the simplex over $\mathcal{A}_i$. We also use notation $\pi_{i,h}(a|s)$ to denote the probability of the $it,(h)$ agent taking action a at state s at step h.

A joint (potentially correlated) policy is a set of policies $\{\pi_i\}_{i=1}^m$, where the same random sample ω is shared among all agents, which we denote as $\pi = \pi_1 \odot \pi_2 \odot \ldots \odot \pi_m$. We also denote $\pi_{-i} = \pi_1 \odot \ldots \pi_{i-1} \odot \pi_{i+1} \odot \ldots \odot \pi_m$ to be the joint policy excluding the $it, (h)$ player. A special case of joint policy is the *product policy* where the random sample has special form $\omega = (\omega_1, \ldots, \omega_m)$, and for any $i \in [m]$, π_i only uses the randomness in ω_i, which is independent of remaining $\{\omega_j\}_{j \neq i}$, which we denote as $\pi = \pi_1 \times \pi_2 \times \ldots \times \pi_m$.

2.1.2 Value

We define the value function $V^{\pi}_{i,1}(s_1)$ as the expected cumulative reward that the $it, (h)$ player will receive if the game starts at initial state s_1 at the 1st step and all players follow joint policy π:

$$V^{\pi}_{i,1}(s_1) := \mathbb{E}_{\pi}\left[\sum_{h=1}^{H} r_{i,h}(s_h, \boldsymbol{a}_h) \middle| s_1\right]. \tag{2.1}$$

where the expectation is taken over the randomness in transition and the random sample ω in policy π.

We also define $Q^{\pi}_{i,h} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to be the Q-value function at step h so that $Q^{\pi}_{i,h}(s, \boldsymbol{a})$ gives the cumulative rewards received under policy π, starting from $(s, \boldsymbol{a})$ at step h:

$$Q^{\pi}_{i,h}(s, \boldsymbol{a}) := \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{i,h'}(s_{h'}, \boldsymbol{a}_{h'}) \middle| s_h = s, \boldsymbol{a}_h = \boldsymbol{a}\right]. \tag{2.2}$$

For simplicity, we define operator $\mathbb{P}_h$ as $[\mathbb{P}_h V](s, \boldsymbol{a}) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,\boldsymbol{a})} V(s')$ for any value function V. We also use notation $[\mathbb{D}_{\pi} Q](s) := \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|s)} Q(s, \boldsymbol{a})$ for any action-value function Q.

By definition of value functions, we have the **Bellman equation**

$$Q^{i,\pi}_{i,h}(s, \boldsymbol{a}) = (r_{i,h} + \mathbb{P}_h V^{\pi}_{i,h+1})(s, \boldsymbol{a}), \qquad V^{\pi}_{i,h}(s) = (\mathbb{D}_{\boldsymbol{a}} Q^{\pi}_{i,h})(s)$$

for all $(s, \boldsymbol{a}, h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H] \times m$, and at the $(H+1)^{\text{th}}$ step we have $V^{\pi}_{i,H+1}(s) = 0$ for all $s \in \mathcal{S}$.

2.1.3 Best response and strategy modification

For any strategy π_{-i}, the *best response* of the $it,(h)$ player is defined as a policy of the $it,(h)$ player which is independent of the randomness in π_{-i}, and achieves the highest value for herself conditioned on all other players deploying π_{-i}. In symbol, the best response is the maximizer of $\max_{\pi_i'} V_{i,1}^{\pi_i' \times \pi_{-i}}(s_1)$ whose value we also denote as $V_{i,1}^{\dagger,\pi_{-i}}(s_1)$ for simplicity. By its definition, we know the best response can always be achieved at *deterministic* policies. We call a policy π_i an ε-best response if

$$\max_{\pi_i'} V_{i,1}^{\pi_i' \times \pi_{-i}}(s_1) - V_{i,1}^{\pi_i \times \pi_{-i}}(s_1) \le \varepsilon. \tag{2.3}$$

A *strategy modification* ϕ_i for the $it,(h)$ player is a set of maps $\phi_i := \{\phi_{i,h} : (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i\}$, where $\phi_{i,h}$ can depend on the history τ_h and maps actions in $\mathcal{A}_i$ to different actions in $\mathcal{A}_i$. For any policy of the $it,(h)$ player π_i, the modified policy (denoted as $\phi_i \diamond \pi_i$) changes the action $\pi_{i,h}(\omega, \tau_h)$ under random sample ω and history τ_h to $\phi_i(\tau_h, \pi_{i,h}(\omega, \tau_h))$.

Here, we only introduce the deterministic strategy modification for simplicity of notation, which is sufficient for discussion in the context of this thesis. The random strategy modification can also be defined by introducing randomness in ϕ_i which is independent of randomness in π_i and π_{-i}. It can be shown that the best strategy modification can always be deterministic.

For any joint policy π, we define the best strategy modification of the $it,(h)$ player as the maximizer of $\max_{\phi_i} V_{i,1}^{(\phi_i \diamond \pi_i) \odot \pi_{-i}}(s_1)$.

Different from the best response, which is completely independent of the randomness in π_{-i}, the best strategy modification changes the policy of the $it,(h)$ player while still utilizing the shared randomness among π_i and π_{-i}. Therefore, the best strategy modification is more powerful than the best response: formally one can show that $\max_{\phi_i} V_{i,1}^{(\phi_i \diamond \pi_i) \odot \pi_{-i}}(s_1) \ge \max_{\pi_i'} V_{i,1}^{\pi_i' \times \pi_{-i}}(s_1)$ for any policy π.

2.2 Equilibria

A special case of Markov game is Markov Decision Process (MDP). One can show there always exists an optimal policy $\pi^\star = \arg\max_\pi V_1^\pi(s_1)$. Denote the value of the optimal policy as $V^\star$. The optimality criteria for a policy in MDPs is to maximize the expected return, or put it another way, minimize the gap between the optimal policy. Quantitatively, we call a policy ε-optimal policy π if $V_1^\star(s_1) - V_1^\pi(s_1) \le \varepsilon$.

The optimality criteria for Markov games is more involved since an optimal policy in the above sense does not necessarily exist. On the countrary, we will consider the notion of **equilibrium**.

For Markov games, there are three common notions of equilibira in the game theory literature—Nash Equilibrium, Correlated Equilibrium and Coarse Correlated Equilibrium.

First, a **Nash equilibrium** is defined as a product policy where no player can increase her value by changing only her own policy. Formally,

Definition 2 (Nash Equilibrium)**.** A *product* policy π is a **Nash equilibrium** if

$$\max_{i\in[m]} (V_{i,1}^{\dagger,\pi_{-i}} - V_{i,1}^{\pi})(s_1) = 0.$$

A *product* policy π is an ε-approximate Nash equilibrium if $\max_{i\in[m]} (V_{i,1}^{\dagger,\pi_{-i}} - V_{i,1}^{\pi})(s_1) \le \varepsilon$.

We remark that, even for a game with **known** parameters $\mathbb{P}(\cdot|\cdot)$ and $\boldsymbol{r}(\cdot)$, the Nash equilibrium in general has been proved PPAD-hard to compute Daskalakis [2013]. As a result, although Nash equilibrium is conceptually the most natural and simple notion, we have to consider other equilibria due to computational feasibility.

Second, a coarse correlated equilibrium is defined as a joint (potentially correlated) policy where no player can increase her value by playing a different independent strategy. In symbol,

Definition 3 (Coarse Correlated Equilibrium)**.** A joint policy π is a **CCE** if

$$\max_{i\in[m]} (V_{i,1}^{\dagger,\pi_{-i}} - V_{i,1}^{\pi})(s_1) = 0.$$

A joint policy π is a ε-approximate CCE if $\max_{i\in[m]} (V_{i,1}^{\dagger,\pi_{-i}} - V_{i,1}^{\pi})(s_1) \le \varepsilon$.

The only difference between Definition 2 and Definition 3 is that Nash equilibrium requires the policy π to be a product policy while CCE does not. Thus, it is clear that CCE is a relaxed notion of Nash equilibrium, and a Nash equilibrium is always a CCE.

Finally, a correlated equilibrium is defined as a joint (potentially correlated) policy where no player can increase her value by using a strategy modification. In symbol,

Definition 4 (Correlated Equilibrium)**.** A joint policy π is a **CE** if

$$\max_{i\in[m]} \max_{\phi_i} (V_{i,1}^{(\phi_i\diamond\pi_i)\odot\pi_{-i}} - V_{i,1}^{\pi})(s_1) = 0.$$

A joint policy π is a ε-approximate CE if $\max_{i\in[m]} \max_{\phi_i} (V_{i,1}^{(\phi_i\diamond\pi_i)\odot\pi_{-i}} - V_{i,1}^{\pi})(s_1) \le \varepsilon$.

In Markov games, we also have that a Nash equilibrium is a CE, and a CE is a CCE.

Proposition 5 (Nash $\subset$ CE $\subset$ CCE)**.** *In Markov games, any ε-approximate Nash equilibrium is an ε-approximate CE, and any ε-approximate CE is an ε-approximate CCE.*

2.3 Two-player Zero-sum games

When can a Nash equilibrium can be computed in polynomial time? The best-known case is Two-player Zero-sum (2p0s) games and as a result it will be discussed separately in many parts of the thesis. We remark 2p0s game is not the only interesting game setting where Nash equilibrium can be computed efficiently. For example, in games where all the players share the same utility function, we can also find a Nash equilibrium in polynoimial time [Song et al., 2022a]. This fully cooperative case is a special case of potential game, but we will not discuss these cases in this thesis. In this thesis, we only present results for finding Nash equilibria in two-player zero-sum MGs.

As the name suggests, in a 2p0s Markov game, the second player's reward is just the first player's negative, i.e., $r_{2,h} = -r_{1,h}$. We call the first player the max-player (with action set $|\mathcal{A}| \le A$) and the second player the min-player (with action set $|\mathcal{B}| \le B$). As a result we can only consider the utility of the max-player and drop the subscript on player (usually using i above) for simplicity.

Technically, to ensure $r_{2,h} \in [0,1]$, we choose $r_{2,h} = 1 - r_{1,h}$. We note that adding a constant to the reward function has no effect on the equilibria. As a result sometimes people also use the term two-player constant-sum to embrace this generality.

2.3.1 Minimiax theorem

The Nash equilibrium can be find efficiently in 2p0s games because there are many important properties that only hold in 2p0s games. We will discuss some of the most significant ones here and leave the more invovled ones later when we cover the algorithmic aspect of MGs.

For any policy of the max-player μ, there exists a *best response* of the min-player, which is a policy $\nu^\dagger(\mu)$ satisfying $V_h^{\mu,\nu^\dagger(\mu)}(s) = \inf_\nu V_h^{\mu,\nu}(s)$ for any $(s,h) \in \mathcal{S} \times [H]$. We denote $V_h^{\mu,\dagger} := V_h^{\mu,\nu^\dagger(\mu)}$. By symmetry, we can also define $\mu^\dagger(\nu)$ and $V_h^{\dagger,\nu}$. It is further known [cf. Filar and Vrieze, 2012] that there exist policies $\mu^\star$, $\nu^\star$ that are optimal against the best responses of the opponents, in the sense that

$$V_h^{\mu^\star,\dagger}(s) = \sup\nolimits_\mu V_h^{\mu,\dagger}(s), \qquad V_h^{\dagger,\nu^\star}(s) = \inf\nolimits_\nu V_h^{\dagger,\nu}(s), \qquad \text{for all } (s,h).$$

These optimal strategies $(\mu^\star, \nu^\star)$ are exactly the Nash equilibria of the Markov game, which satisfies the following minimax equation

$$\sup\nolimits_\mu \inf\nolimits_\nu V_h^{\mu,\nu}(s) = V_h^{\mu^\star,\nu^\star}(s) = \inf\nolimits_\nu \sup\nolimits_\mu V_h^{\mu,\nu}(s).$$

Notice the minimax theorem here is different from the one for matrix games, i.e. $\max_\phi \min_\psi \phi^\top A\psi = \min_\psi \max_\phi \phi^\top A\psi$ for any matrix A, since here $V_h^{\mu,\nu}(s)$ is in general not bilinear in μ, ν.

We further abbreviate the values of Nash equilibrium $V_h^{\mu^\star,\nu^\star}$ and $Q_h^{\mu^\star,\nu^\star}$ as $V_h^\star$ and $Q_h^\star$.

In 2p0s games we have a better way to quantify the optimality of a pair of policy $(\widehat{\mu}, \widehat{\nu})$ using the gap between their performance and the performance of the optimal strategy (i.e., Nash equilibrium) when playing against the best responses respectively:

$$V_1^{\dagger,\widehat{\nu}}(s_1) - V_1^{\widehat{\mu},\dagger}(s_1) = \Big[V_1^{\dagger,\widehat{\nu}}(s_1) - V_1^\star(s_1)\Big] + \Big[V_1^\star(s_1) - V_1^{\widehat{\mu},\dagger}(s_1)\Big].$$

A pair of general policies $(\widehat{\mu}, \widehat{\nu})$ is an **ε-approximate Nash equilibrium**, if $V_1^{\dagger,\widehat{\nu}}(s_1) - V_1^{\widehat{\mu},\dagger}(s_1) \le \varepsilon$.

The existence of the value of a 2p0s games makes it possible to compute Nash Equilibrium through CCEs. To describe this result we first restate the definition of CCE (Definition 3) after rescaling. For any pair of matrices $P, Q \in [0,1]^{n\times m}$, the subroutine $\mathrm{CCE}(P, Q)$ returns a distribution $\pi \in \Delta_{n\times m}$ that satisfies:

$$\mathbb{E}_{(a,b)\sim\pi} P(a,b) \ge \max_{a^\star} \mathbb{E}_{(a,b)\sim\pi} P(a^\star, b), \quad \mathbb{E}_{(a,b)\sim\pi} Q(a,b) \le \min_{b^\star} \mathbb{E}_{(a,b)\sim\pi} Q(a, b^\star) \tag{2.4}$$

We make three remarks on CCE. First, a CCE always exists since a Nash equilibrium for a general-sum game with payoff matrices (P, Q) is also a CCE defined by (P, Q), and a Nash equilibrium always exists. Second, a CCE can be efficiently computed, since above constraints (2.4) for CCE can be rewritten as $n + m$ linear constraints on $\pi \in \Delta_{n\times m}$, which can be efficiently resolved by standard linear programming algorithm. Third, a CCE in general-sum games needs not to be a Nash equilibrium. However, a CCE in zero-sum games is guaranteed to be a Nash equalibrium.

Proposition 6. *Let $\pi = \mathrm{CCE}(Q, Q)$, and (μ, ν) be the marginal distribution over both players' actions induced by π. Then (μ, ν) is a Nash equilibrium for payoff matrix Q.*

Intuitively, a CCE procedure can be used in Nash Q-learning for finding an approximate Nash equilibrium, because the values of upper confidence and lower confidence ($\overline{Q}$ and $\underline{Q}$) will be eventually very close, so that the preconditions of Proposition 6 becomes approximately satisfied.

2.3.2 Bellman optimality equations

Now we are ready to present the Bellman optimality equations for the value functions of the best responses and the Nash equilibrium.

Best responses. For any Markov policy μ of the max-player, by definition, we have the following Bellman equations for values of its best response:

$$Q_h^{\mu,\dagger}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{\mu,\dagger})(s,a,b), \qquad V_h^{\mu,\dagger}(s) = \inf_{\nu \in \Delta_{\mathcal{B}}} (\mathbb{D}_{\mu_h \times \nu} Q_h^{\mu,\dagger})(s),$$

for all $(s,a,b,h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H]$, where $V_{H+1}^{\mu,\dagger}(s) = 0$ for all $s \in \mathcal{S}$.

Similarly, for any Markov policy ν of the min-player, we also have the following symmetric version of Bellman equations for values of its best response:

$$Q_h^{\dagger,\nu}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{\dagger,\nu})(s,a,b), \qquad V_h^{\dagger,\nu}(s) = \sup_{\mu \in \Delta_{\mathcal{A}}} (\mathbb{D}_{\mu \times \nu_h} Q_h^{\dagger,\nu})(s).$$

for all $(s,a,b,h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H]$, where $V_{H+1}^{\dagger,\nu}(s) = 0$ for all $s \in \mathcal{S}$.

Nash equilibria. Finally, by definition of Nash equilibria in Markov games, we have the following Bellman optimality equations:

$$\begin{aligned} Q_h^{\star}(s,a,b) =& (r_h + \mathbb{P}_h V_{h+1}^{\star})(s,a,b) \\ V_h^{\star}(s) = \sup_{\mu \in \Delta_{\mathcal{A}}} & \inf_{\nu \in \Delta_{\mathcal{B}}} (\mathbb{D}_{\mu \times \nu} Q_h^{\star})(s) = \inf_{\nu \in \Delta_{\mathcal{B}}} \sup_{\mu \in \Delta_{\mathcal{A}}} (\mathbb{D}_{\mu \times \nu} Q_h^{\star})(s) \end{aligned}$$

for all $(s,a,b,h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H]$, where $V_{H+1}^{\star}(s) = 0$ for all $s \in \mathcal{S}$.

2.4 Learning objectives

How do we measure the performance of a learning algorithm in Markov games? In this section we will cover the two most common learning objectives: regret minimization and sample complexity (of learning an equilibrium).

Online learning The most natural idea to measure the performance of a learning algorithm is to compare with the best response to the min-player's policy in average. This motivates the defition of regret competing against the best fixed policy in hindsight:

$$\text{Regret}(K) := \sup_{\mu} \sum_{k=1}^{K} \big(V_1^{\mu,\nu^k}(s_1) - V_1^{\mu^k,\nu^k}(s_1)\big), \tag{2.5}$$

where the superscript k denotes the corresponding objects in the k-th episode. Although we use this compact notation, the regret depends on both μ^k and ν^k.

In the **online learning** setting, our goal is to maximize the expected cumulative return, or equivalently, to minimize the regret. We call a learning algorithm if the regret grows sublinearly in K, as the average regret will converge to zero in long run.

Notice this criteria is purely unilateral and imposes no assumption on the min-player. Also it measures the performance in K consecutive episodes instead of a single policy.

Self-play Another common of reinforcement learning is to design algorithms for Markov games that can find an ε-approximate Nash equilibrium, CE or CCE using a number of episodes that is small in its dependency on S, A, B, H as well as $1/\varepsilon$. An upper bound of this number of episodes is known as PAC sample complexity bound, or sample complexity for short.

The self-play setting is closely related to the learning agents trained through comepeting with each other (For example, Silver et al. [2016]). Here we can control all of the agents instead of just one of them. Different from the online learning setting, we no longer care about the return in the training process but only the output policy at the end of the training process, which is not totally reasonable but simplifies the formulation greatly.

Online-to-batch conversion Interestingly, the two notions introduced above are closely related. In MDPs, any sublinear regret algorithm can be directly converted to a polynomial-sample PAC algorithm via the standard online-to-batch conversion. For example, see Jin et al. [2018].

For games, the intuition remains valid but things become more technical. Roughly speaking, if we have a sublinear regret bound, it will translate into a polynomial-sample PAC bound for learning CCE in general games or NE in 2p0s games. For CE, we need stronger notion of regret defined through policy modifications. We will present the formal reductions later when needed.

2.5 Hardness results

The most common approach to establish PAC-learning results for MDP and matrix games is to first control the regret and then use the online-to-batch conversion as we introduced in Section 2.4. However, this approach does not work for Markov games. In this section, we present two hardness results for achieving sublinear regret in 2p0s Markov games when playing against adversarial opponents, which rules out a popular approach to design algorithms for finding Nash equilibria.

2.5.1 Computational hardness

We begin with a computational hardness result, showing computing the best response against an opponent with a fixed unknown policy is hard. This will imply achieving sublinear regret in Markov games when playing against adversarial opponents is also computationally hard.

Computing the best response We say an algorithm is a *polynomial time algorithm for learning the best response* if for any policy of the opponent ν, and for any

$\varepsilon > 0$, the algorithm finds the ε-approximate best response of policy ν (Defined in Equation (2.3)) with probability at least $1/2$, in time polynomial in $S, H, A, B, \varepsilon^{-1}$.

We can show the following hardness result for finding the best response in polynomial time.

Theorem 7 (Hardness for learning the best response)**.** *There exists a Markov game with deterministic transitions and rewards defined for any horizon $H \geq 1$ with $S = 2$, $A = 2$, and $B = 2$, such that if there exists a polynomial time algorithm for learning the best response for this Markov game, then there exists a polynomial time algorithm for learning parity with noise (see problem description in Appendix A.3).*

We remark that learning parity with noise is a notoriously difficult problem that has been used to design efficient cryptographic schemes. It is conjectured by the community to be hard.

Conjecture 8 (Kearns [1998])**.** *There is no polynomial time algorithm for learning party with noise.*

Theorem 7 with Conjecture 8 demonstrates the fundamental difficulty—if not strict impossibility—of designing a polynomial time for learning the best responses in Markov games.

The intuitive reason for such computational hardness is that, while the underlying system has Markov transitions, the opponent can play policies that encode long-term correlations with non-Markovian nature, such as parity with noise, which makes it very challenging to find the best response. It is known that learning many other sequential models with long-term correlations (such as hidden Markov models or partially observable MDPs) is as hard as learning parity with noise Mossel and Roch [2005].

Actually, if the opponent is restricted to only play Markov policies, then learning the best response is as easy as learning a optimal policy in the standard single-agent Markov decision process, where efficient algorithms are known to exist. Nevertheless, when the opponent can as well play any policy which may be non-Markovian, as shown in Theorem 7, finding the best response against those policies is computationally challenging.

Playing Against Adversarial Opponent Theorem 7 directly implies the difficulty for achieving sublinear regret in Markov games when playing against adversarial opponents in Markov games. Our construction of hard instances in the proof of Theorem 7 further allows the adversarial opponent to only play Markov policies in each episode.

Since playing against adversarial opponent is a different problem with independent interest, we present the full result here.

Without loss of generality, we still consider the setting where the algorithm can only control the max-player, while the min-player is an adversarial opponent. In the beginning of every episode k, both players pick their own policies μ^k and ν^k, and execute them throughout the episode. The adversarial opponent can possibly pick her policy ν^k *adaptive* to all the observations in the earlier episodes.

We say an algorithm for the learner is a *polynomial time no-regret algorithm* if there exists a $\delta > 0$ such that for *any* adversarial opponent, and any fixed $K > 0$, the algorithm outputs policies $\{\mu^k\}_{k=1}^K$ which satisfies the following, with probability at least 1/2, in time polynomial in S, H, A, B, K.

$$\mathfrak{R}(K) = \sup_{\mu} \sum_{k=1}^{K} V_1^{\mu,\nu^k}(s_1) - \sum_{k=1}^{K} V_1^{\mu^k,\nu^k}(s_1) \leq \mathrm{poly}(S, H, A, B) K^{1-\delta} \tag{2.6}$$

Theorem 7 directly implies the following hardness result for achieving no-regret against adversarial opponents, a result that first appeared in [Radanovic et al., 2019].

Corollary 9 (Hardness for playing against adversarial opponent)**.** *There exists a Markov game with deterministic transitions and rewards defined for any horizon $H \geq 1$ with $S = 2$, $A = 2$, and $B = 2$, such that if there exists a polynomial time no-regret algorithm for this Markov game, then there exists a polynomial time algorithm for learning parity with noise (see problem description in Appendix A.3). The claim remains to hold even if we restrict the adversarial opponents in the Markov game to* *be non-adaptive, and to only play Markov policies in each episode.*

Similar to Theorem 7, Corollary 9 combined with Conjecture 8 demonstrates the fundamental difficulty of designing a polynomial time no-regret algorithm against adversarial opponents for Markov games.

2.5.2 Statistical hardness in unknown games

In unknown Markov games, where each agent cannot access the other agents' actions, we can establish a stronger statistical hardness result. In this section, we show that, competing against the best policy in hindsight is statistically intractable in general. In particular, we show that in this case, the regret has to be either linear in K or exponential in H.

Theorem 10 (Statistical hardness for online learning in unknown MGs)**.** *For any $H \geq 2$ and $K \geq 1$, there exists a two-player zero-sum MG with horizon H, $|S| \leq 2$, $|A| \leq 2$, $|B| \leq 4$ such that any algorithm for unknown MGs suffers the following worst-case one-sided regret:*

$$\sup_{\mu} \sum\nolimits_{k=1}^{K} \Big(V_1^{\mu,\nu^k}(s_1) - \mathbb{E}_{\mu^k} V_1^{\mu^k,\nu^k}(s_1) \Big) \geq \Omega\big(\min\{\sqrt{2^H K}, K\}\big).$$

In particular, any algorithm has to suffer linear regret unless $K \geq \Omega(2^H)$.

Proof Sketch We start by considering online learning in (single-agent) MDPs, where the reward and transition function in each episode are adversarially determined, and the goal is to compete against the best (fixed) policy in hindsight. In the following lemma we show that this problem is statistically hard; see Lemma 64 in the Appendix A.4 for its formal statement.

Lemma 11. *(informal) For any algorithm, there exists a sequence of single agent MDPs with horizon H, $S = O(H)$ states and $A = O(1)$ actions, such that the regret defined against the best policy in hindsight is $\Omega(\min\{\sqrt{2^H K}, K\})$.*

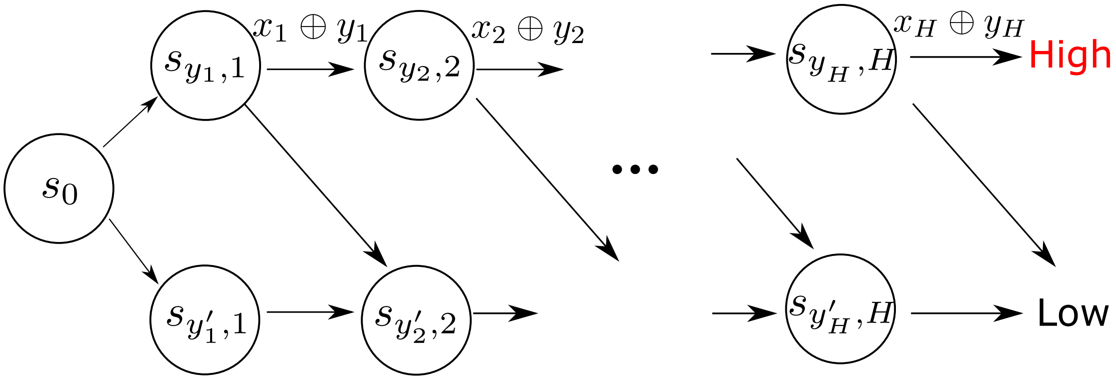


Figure 2-2: Illustration of the MDP $M_{X,Y}$. For $y \in \{0,1\}$, y' stands for $1-y$.

We now briefly explain how this family of hard MDPs is constructed, which is inspired by the "combination lock" MDP [Du et al., 2019]. Every MDP $M_{X,Y}$ is specified by two H-bit strings: $X, Y \in \{0,1\}^H$. The states are $\{s_{0,0}, s_{0,1}, s_{1,1}, \cdots, s_{0,H}, s_{1,H}\}$. As shown in Figure 2-2, $M_{X,Y}$ has a layered structure, and the reward is nonzero only at the final layer. The only way to achieve the high reward is to follow the path $s_{0,0} \to s_{y_1,1} \to \cdots s_{y_H,H}$. Thus, the corresponding optimal policy is $\pi(s_{w,h}) = x_h \oplus w$, which is only a function of X. Here, $\oplus$ denotes the bitwise exclusive or operator.

Now, in each episode, Y is chosen from a uniform distribution over $\{0,1\}^H$ while X is fixed. When the player interacts with $M_{X,Y}$, since Y is uniformly random, it gets no effective feedback from the observed transitions, and the only informative feedback is the reward at the end. However, achieving the high reward requires guessing every bit of X correctly. This "needle in a haystack" situation makes the problem as hard as a multi-armed bandit problem with 2^H arms. The regret lower bound immediately follows.

Next, we use the hard family of MDPs in Lemma 64 to prove Theorem 10 by reducing the adversarial MDP problem to online learning in unknown MGs. The construction is straightforward. The state space and the action space for the max-player are the same as that in the original MDP family. The min-player has control over the transition function and reward at each step, and executes a policy such that the induced MDP for the max-player is the same as $M_{X,Y}$. This is possible using only $B = O(1)$ actions as $M_{X,Y}$ has a layered structure. Online learning in unknown MGs then simulates the online learning in the adversarial MDP problem, and thus has the same regret lower bound.

Chapter 3

Markov Games: Model-based Learning

One prevalent approach towards solving multi-agent RL is *model-based* methods, that is, to use the existing visitation data to build an estimate of the model (i.e. transition dynamics and rewards), run an offline planning algorithm on the estimated model to obtain the policy, and play the policy in the environment. Such a principle underlies some of the earliest single-agent online RL algorithms such as E3 [Kearns and Singh, 2002] and RMax [Brafman and Tennenholtz, 2002], and is conceptually appealing for multi-agent RL too since the multi-agent structure does not add complexity onto the model estimation part and only requires an appropriate multi-agent planning algorithm (such as value iteration for games [Shapley, 1953]) in a black-box fashion. On the other hand, *model-free* methods do not directly build estimates of the model, but instead directly estimate the value functions or action-value (Q) functions of the problem at the optimal/equilibrium policies, and play the greedy policies with respect to the estimated value functions. Model-free algorithms have also been well developed for multi-agent RL such as friend-or-foe Q-Learning [Littman, 2001] and Nash Q-Learning [Hu and Wellman, 2003].

While both model-based and model-free algorithms have been shown to be provably efficient in multi-agent RL in a recent line of work [Bai and Jin, 2020, Xie et al., 2020, Bai et al., 2020], a more precise understanding of the optimal sample complexities within these two types of algorithms (respectively) is still lacking. In the specific setting of two-player zero-sum Markov games, the current best sample complexity for model-based algorithms is achieved by the VI-ULCB (Value Iteration with Upper/Lower Confidence Bounds) algorithm [Bai and Jin, 2020, Xie et al., 2020]: In a tabular Markov game with S states, $\{A, B\}$ actions for the two players, and horizon length H, VI-ULCB is able to find an ε-approximate Nash equilibrium policy in $\tilde{\mathcal{O}}(H^4S^2AB/\varepsilon^2)$ episodes of game playing. However, compared with the information-theoretic lower bound $\Omega(H^3S(A+B)/\varepsilon^2)$, this rate has suboptimal dependencies on all of H, S, and A, B. In contrast, the current best sample complexity for *model-free* algorithms is achieved by Nash V-Learning [Bai et al., 2020], which finds an ε-approximate Nash policy in $\tilde{\mathcal{O}}(H^6S(A+B)/\varepsilon^2)$ episodes. Compared with the lower bound, this is tight except for a poly(H) factor, which may seemingly suggest that model-free algorithms could be superior to model-based ones in multi-agent RL. However, such a conclusion would be in stark contrast to the single-agent MDP setting, where it is known that model-based algorithms are able to achieve minimax optimal sample complexities [Jaksch et al., 2010, Azar et al., 2017]. It naturally arises whether model-free algorithms are indeed superior in multi-agent settings, or whether the existing analyses of model-based algorithms are not tight. This motivates us to ask the following question:

How *sample-efficient* are *model-based* algorithms in multi-agent RL?

In this chapter, we advance the theoretical understandings of multi-agent RL by presenting a sharp analysis of model-based algorithms on Markov games. Our core contribution is the design of a new model-based algorithm *Optimistic Nash Value Iteration* (Nash-VI) that achieves an almost optimal sample complexity for zero-sum Markov games and improves significantly over existing model-based approaches.

3.1 Player coordination

The lower bound presented in Section 2.5.1 and Section 2.5.2 rule out the possiblity of controlling the sampling complexity through regret (defined in Equation (2.5)) minimization, by designing an *independent* learning algorithm for each of the players. In this section, we present the first workaround, by taking a model-based approach. Concretely, we coordinate the players through a centralized model estimator.

The main difference between independent learning and player coordination is : Independent learning is actually an overshoot because using the Nash Folklore theorem itself does not require the learning dynamics for each of the player is independent. Although independent learning has its own advantages, we have seen achieving low regret through independent learning is provably hard. As long as we have a smart way to coordinate the players and prove under such coordination the regret can be controlled, we can essentially achieve the same goal.

How to achieve such coordination? Model-based approach is the most ideal choice. Although we do not know the underlying MG, we can estimate the state transition probabilities through samples and provide an estimated MG. Then we can use a centralized planner to find the equilibrium of this estimated MG and if the estimation is good enough, the regret for all the players must be low because the their policy should be already very close to a Nash equilibrium.

To make the presentation simpler, we will use a new and stronger form of regret which is more suitable for model-based learning. Through the classical online-to-batch conversion, the regret guarantees can again be translated into sample complexity results that we are looking for. The definition of regret for 2p0s games used throughout this chapter is given below, which we will denote by Nash regret.

Definition 12 (Nash Regret in 2p0s games)**.** Let (μ^k, ν^k) denote the policies deployed by the algorithm in the k^{th} episode. After a total of K episodes, the regret is defined as

$$\mathfrak{R}(K) = \sum_{k=1}^{K} (V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1).$$

Similarly we can define the counterparts for general games as below.

Definition 13 (Nash-regret in general-sum MGs)**.** Let π^k denote the (product) policy deployed by the algorithm in the k^{th} episode. After a total of K episodes, the regret is defined as

$$\mathfrak{R}_{\mathsf{Nash}}(K) = \sum_{k=1}^{K} \max_{i\in[m]} (V_{1,i}^{\dagger,\pi_{-i}^{k}} - V_{1,i}^{\pi^k})(s_1).$$

Definition 14 (CCE-regret in general-sum MGs)**.** Let policy π^k denote the (correlated) policy deployed by the algorithm in the k^{th} episode. After a total of K episodes, the regret is defined as

$$\mathfrak{R}_{\mathsf{CCE}}(K) = \sum_{k=1}^{K} \max_{i\in[m]} (V_{1,i}^{\dagger,\pi_{-i}^{k}} - V_{1,i}^{\pi^k})(s_1).$$

Definition 15 (CE-regret in multiplayer Markov games)**.** Let policy π^k denote the policy deployed by the algorithm in the k^{th} episode. After a total of K episodes, the regret is defined as

$$\mathfrak{R}_{\mathsf{CE}}(K) = \sum_{k=1}^{K} \max_{i\in[m]} \max_{\phi\in\Phi_i} (\ V_{1,i}^{\phi\diamond\pi^k} - V_{1,i}^{\pi^k})(s_1).$$

To make the exposition more accessible, we will first present the algorithms and theoretical guarantees for **two-player zero-sum games**, where the presentation is relatively simpler, and then consider general games.

3.2 Optimistic Nash Value Iteration

In this section, we present our main algorithm—Optimistic Nash Value Iteration (Nash-VI), and provide its theoretical guarantee.

3.2.1 Algorithm description

We describe our Nash-VI Algorithm 1. In each episode, the algorithm can be decomposed into two parts.

- Line 3-15 (Optimistic planning from the estimated model): Performs value iteration with bonus using the empirical estimate of the transition $\widehat{\mathbb{P}}$, and computes a new (joint) policy π which is “greedy” with respect to the estimated value functions;

- Line 18-21 (Play the policy and update the model estimate): Executes the policy π, collects samples, and updates the estimate of the transition $\widehat{\mathbb{P}}$.

At a high-level, this two-phase strategy is standard in the majority of model-based RL algorithms, and also underlies provably efficient model-based algorithms such as UCBVI for single-agent (MDP) setting [Azar et al., 2017] and VI-ULCB for the two-player Markov game setting [Bai and Jin, 2020]. However, VI-ULCB has two undesirable drawbacks: the sample complexity is not tight in any of H, S, and A, B dependency, and its computational complexity is PPAD-complete (a complexity class conjectured to be computationally hard [Daskalakis, 2013]).

As we elaborate in the following, our Nash-VI algorithm differs from VI-ULCB in a few important technical aspects, which allows it to significantly improve the sample complexity over VI-ULCB, and ensures that our algorithm terminates in polynomial time.

Before digging into explanations of techniques, we remark that line 16-17 is only used for computing the output policies. It chooses policy π^{out} to be the policy in the episode with minimum gap $(\overline{V}_1 - \underline{V}_1)(s_1)$. Our final output policies $(\mu^{\text{out}}, \nu^{\text{out}})$ are simply the *marginal policies* of π^{out}. That is, for all $(s, h) \in \mathcal{S} \times [H]$, $\mu_h^{\text{out}}(\cdot|s) := \sum_{b \in \mathcal{B}} \pi_h^{\text{out}}(\cdot, b|s)$, and $\nu_h^{\text{out}}(\cdot|s) := \sum_{a \in \mathcal{A}} \pi_h^{\text{out}}(a, \cdot|s)$.

3.2.2 Overview of techniques

Auxiliary bonus γ. The major improvement over VI-ULCB [Bai and Jin, 2020] comes from the use of a different style of bonus term γ (line 9), in addition to the standard bonus β (line 8), in value iteration steps (line 10-11). See Figure 3-1 for an illustration.

This is also the main technical contribution of our Nash-VI algorithm. This auxiliary bonus γ is computed by applying the empirical transition matrix $\widehat{\mathbb{P}}_h$ to the gap

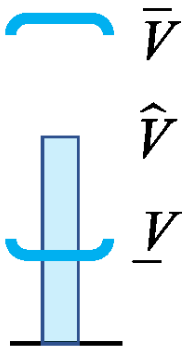


Figure 3-1: A graphical view of the upper and lower bounds.

at the next step $\overline{V}_{h+1} - \underline{V}_{h+1}$, This is very different from standard bonus β, which is typically designed according to the concentration inequalities.

The main purpose of these value iteration steps (line 10-11) is to ensure that the estimated values $\overline{Q}_h$ and $\underline{Q}_h$ are with high probability the upper bound and the lower bound of the Q-value of the current policy when facing best responses (see Lemma 68 and 70 for more details). To do so, prior work [Bai and Jin, 2020] only adds bonus β, which needs to be as large as $\tilde{\Theta}(\sqrt{S/t})$. In contrast, the inclusion of auxiliary bonus γ in our algorithm allows a much smaller choice for bonus β—which scales only as $\tilde{\mathcal{O}}(\sqrt{1/t})$—while still maintaining valid confidence bounds. This technique alone brings down the sample complexity to $\tilde{\mathcal{O}}(H^4SAB/\varepsilon^2)$, removing an entire S factor compared to VI-ULCB. Furthermore, the coefficient in γ is only c/H for some absolute constant c, which ensures that the introduction of error term γ would hurt the overall sample complexity only up to a constant factor. See Figure 3-2 for an illustration.

We remark that the current policy is stochastic. This is different from the single-agent setting, where the algorithm only seeks to provide an upper bound of the value of the optimal policy where the optimal policy is not random. Due to this difference, the techniques of Azar et al. [2017] cannot be directly applied here.

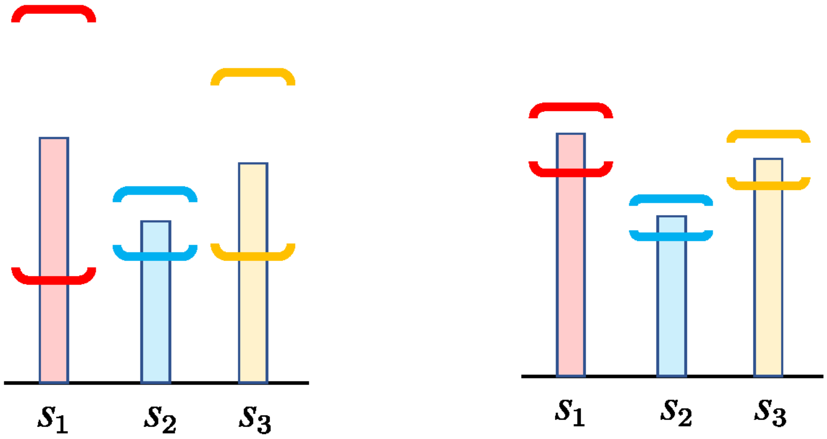


Figure 3-2: Insight I: Sharper confidence bound in the next step induces less uncertainty. The value function estimations are the same for both instances, but the left instance has higher uncertainty in the next step so we would expect the uncertinty is also higher in the current step.

Bernstein concentration. Our Nash-VI allows two choices of the bonus function $\beta = \textsc{Bonus}(t, \widehat{\sigma}^2)$:

$$\begin{cases} \text{Hoeffding type:} \quad c(\sqrt{H^2\iota/t} + H^2S\iota/t), \\ \text{Bernstein type:} \quad c(\sqrt{\widehat{\sigma}^2\iota/t} + H^2S\iota/t), \end{cases} \tag{3.1}$$

where $\widehat{\sigma}^2$ is the estimated variance, ι is the logarithmic factors and c is absolute constant. The $\widehat{\mathbb{V}}$ in line 8 is the empirical variance operator defined as $\widehat{\mathbb{V}}_h V = \widehat{\mathbb{P}}_h V^2 - (\widehat{\mathbb{P}}_h V)^2$ for any $V \in [0, H]^S$. The design of both bonuses stem from the Hoeffding and Bernstein concentration inequalities. Further, the Bernstein bonus uses a sharper concentration, which saves an H factor in sample complexity compared to the Hoeffding bonus (similar to the single-agent setting [Azar et al., 2017]). This further reduces the sample complexity to $\tilde{\mathcal{O}}(H^3SAB/\varepsilon^2)$ which matches the lower bound in all H, S, ε factors. See Figure 3-3 for an illustration.

Coarse Correlated Equilibrium (CCE). The prior algorithm VI-ULCB [Bai and Jin, 2020] computes the "greedy" policy with respect to the estimated value functions by directly computing the Nash equilibrium for the Q-value at each step h. However, since the algorithm maintains both the upper confidence bound and

Algorithm 1 Optimistic Nash Value Iteration (Nash-VI)

```
Initialize: for any $(s, a, b, h)$, $\overline{Q}_h(s, a, b) \leftarrow H$,
  $\underline{Q}_h(s, a, b) \leftarrow 0$, $\Delta \leftarrow H$, $N_h(s, a, b) \leftarrow 0$.
for episode $k = 1, \ldots, K$ do
  for step $h = H, H-1, \ldots, 1$ do
    for $(s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ do
      $t \leftarrow N_h(s, a, b)$.
      if $t > 0$ then
        $U_{h+1} \leftarrow (\overline{V}_{h+1} + \underline{V}_{h+1})/2$.
        $\beta \leftarrow \textsc{Bonus}(t, \widehat{\mathbb{V}}_h[U_{h+1}](s, a, b))$.
        $\gamma \leftarrow (c/H)\widehat{\mathbb{P}}_h(\overline{V}_{h+1} - \underline{V}_{h+1})(s, a, b)$.
        $\overline{Q}_h(s, a, b) \leftarrow \min\{(r_h + \widehat{\mathbb{P}}_h\overline{V}_{h+1})(s, a, b) + \gamma + \beta, H\}$.
        $\underline{Q}_h(s, a, b) \leftarrow \max\{(r_h + \widehat{\mathbb{P}}_h\underline{V}_{h+1})(s, a, b) - \gamma - \beta, 0\}$.
    for $s \in \mathcal{S}$ do
      $\pi_h(\cdot, \cdot|s) \leftarrow \mathrm{CCE}(\overline{Q}_h(s, \cdot, \cdot), \underline{Q}_h(s, \cdot, \cdot))$.
      $\overline{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h}\overline{Q}_h)(s)$.
      $\underline{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h}\underline{Q}_h)(s)$.
  if $(\overline{V}_1 - \underline{V}_1)(s_1) < \Delta$ then
    $\Delta \leftarrow (\overline{V}_1 - \underline{V}_1)(s_1)$ and $\pi^{\mathrm{out}} \leftarrow \pi$.
  for step $h = 1, \ldots, H$ do
    Take action $(a_h, b_h) \sim \pi_h(\cdot, \cdot|s_h)$, observe reward $r_h$ and next state $s_{h+1}$.
    Add 1 to $N_h(s_h, a_h, b_h)$ and $N_h(s_h, a_h, b_h, s_{h+1})$.
    $\widehat{\mathbb{P}}_h(\cdot|s_h, a_h, b_h) \leftarrow N_h(s_h, a_h, b_h, \cdot)/N_h(s_h, a_h, b_h)$.
Output the marginal policies of $\pi^{\mathrm{out}}$: $(\mu^{\mathrm{out}}, \nu^{\mathrm{out}})$.
```

lower confidence bound of the Q-value, this leads to the requirement to compute the Nash equilibrium for a two-player general-sum matrix game, which is in general PPAD-complete [Daskalakis, 2013].

To overcome this computational challenge, we compute a relaxation of the Nash equilibrium—*Coarse Correlated Equalibirum (CCE)*—instead, a technique first introduced by Xie et al. [2020] to address reinforcement learning problems in Markov Games. Formally, for any pair of matrices $\overline{Q}, \underline{Q} \in [0, H]^{A \times B}$, $\mathrm{CCE}(\overline{Q}, \underline{Q})$ returns a distribution $\pi \in \Delta_{\mathcal{A} \times \mathcal{B}}$ such that

$$\begin{cases} \mathbb{E}_{(a,b)\sim\pi}\overline{Q}(a, b) \geq \max_{a^\star} \mathbb{E}_{(a,b)\sim\pi}\overline{Q}(a^\star, b), \\ \mathbb{E}_{(a,b)\sim\pi}\underline{Q}(a, b) \leq \min_{b^\star} \mathbb{E}_{(a,b)\sim\pi}\underline{Q}(a, b^\star). \end{cases} \tag{3.2}$$

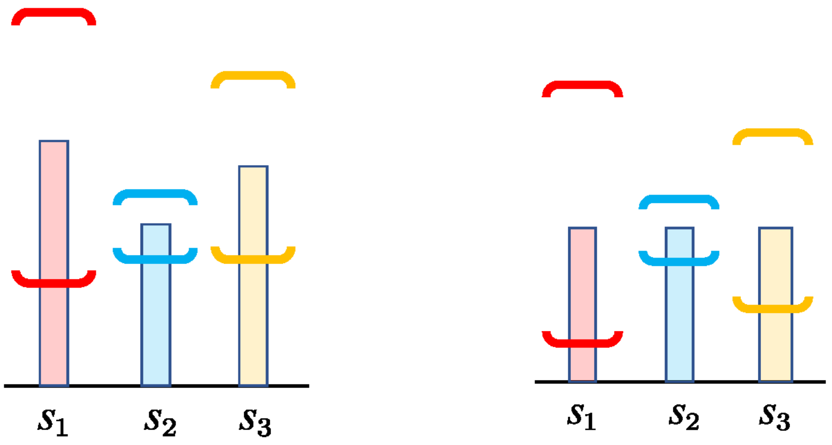


Figure 3-3: Insight II: Lower variance of estimations induces less uncertainty. The value length of confidence bounds are the same for both instances, but the left instance has higher variance in value function estimation in the next step so we would expect the uncertinty is also higher in the current step.

Intuitively, in a CCE the players choose their actions in a potentially correlated way such that no one can benefit from unilateral unconditional deviation. A CCE always exists, since Nash equilibrium is also a CCE and a Nash equilibrium always exists. Furthermore, a CCE can be computed by linear programming in polynomial time. We remark that different from Nash equilibrium where the policies of each player are independent, the policies given by CCE are in general correlated for each player. Therefore, executing such a policy (line 19) requires the cooperation of two players.

3.2.3 Theoretical guarantees

Now we are ready to present the theoretical guarantees for Algorithm 1. We let π^k denote the policy computed in line 13 in the k^{th} episode, and μ^k, ν^k denote the *marginal policy* of π^k for each player.

Theorem 16 (Nash-VI with Hoeffding bonus). *For any $p \in (0, 1]$, letting $\iota = \log(SABT/p)$, then with probability at least $1 - p$, Algorithm 1 with Hoeffding type bonus* (3.1) *(with some absolute $c > 0$) achieves:*

- *The output policies (μ^{out}, ν^{out}) satisfy $(V_1^{\dagger,\nu^{out}} - V_1^{\mu^{out},\dagger})(s_1) \le \varepsilon$ if we choose*

$$K \ge \Omega\left(\frac{H^4 SAB\iota}{\varepsilon^2} + \frac{H^3 S^2 AB\iota^2}{\varepsilon}\right).$$

- *The algorithm has regret bound*

$$\mathfrak{R}(K) = \sum_{k=1}^{K} (V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1) \le \mathcal{O}(\sqrt{H^3 SABT\iota} + H^3 S^2 AB\iota^2),$$

  where $T = KH$ is the total number of steps played within K episodes.

Theorem 16 provides both a sample complexity bound and a regret bound for Nash-VI to find an ε-approximate Nash equilibrium. For small $\varepsilon \le H/(S\iota)$, the sample complexity scales as $\tilde{\mathcal{O}}(H^4 SAB/\varepsilon^2)$. Similarly, for large $T \ge H^3 S^3 AB\iota^3$, the regret scales as $\tilde{\mathcal{O}}(\sqrt{H^3 SABT})$. Theorem 16 is significant in that it improves the sample complexity of the model-based algorithm in Markov games from S^2 to S (and the regret from S to $\sqrt{S}$). This is achieved by adding the new auxiliary bonus γ in value iteration steps as explained in Section 3.2.1. The proof of Theorem 16 can be found in Appendix B.1.1.

Our next theorem states that when using Bernstein bonus instead of Hoeffding bonus as in (3.1), the sample complexity of Nash-VI algorithm can be further improved by a H factor in the leading order term (and the regret improved by a $\sqrt{H}$ factor).

Theorem 17 (Nash-VI with the Bernstein bonus)**.** *For any $p \in (0, 1]$, letting $\iota = \log(SABT/p)$, then with probability at least $1 - p$, Algorithm 1 with Bernstein type bonus* (3.1) *(with some absolute $c > 0$) achieves:*

- *The output policies (μ^{out}, ν^{out}) satisfy $(V_1^{\dagger,\nu^{out}} - V_1^{\mu^{out},\dagger})(s_1) \le \varepsilon$ if we choose*

$$K \ge \Omega\left(\frac{H^3 SAB\iota}{\varepsilon^2} + \frac{H^3 S^2 AB\iota^2}{\varepsilon}\right).$$

- *The algorithm has regret bound*

$$\mathfrak{R}(K) = \sum_{k=1}^{K} (V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1) \le \mathcal{O}(\sqrt{H^2SABT\iota} + H^3S^2AB\iota^2),$$

where $T = KH$ is the total number of steps played within K episodes.

Compared with the information-theoretic sample complexity lower bound $\Omega(H^3S(A+B)\iota/\varepsilon^2)$ and regret lower bound $\Omega(\sqrt{H^2S(A+B)T})$ [Bai and Jin, 2020], when ε is small, Nash-VI with Bernstein bonus achieves the optimal dependency on all of H, S, ε up to logarithmic factors in both the sample complexity and the regret, and the only gap that remains open is a $AB/(A+B) \le \min\{A, B\}$ factor. The proof of Theorem 17 can be found in Appendix B.1.2.

3.3 Reward-free Learning

In this section, we modify our model-based algorithm Nash-VI for the reward-free exploration setting.

Formally, reward-free learning has two phases: In the exploration phase, the agent collects a dataset of transitions $\mathcal{D} = \{(s_{k,h}, a_{k,h}, b_{k,h}, s_{k,h+1})\}_{(k,h)\in[K]\times[H]}$ from a Markov game $\mathcal{M}$ without the guidance of reward information. After the exploration, in the planning phase, for each task $i \in [N]$, $\mathcal{D}$ is augmented with stochastic reward information to become $\mathcal{D}^i = \{(s_{k,h}, a_{k,h}, b_{k,h}, s_{k,h+1}, r_{k,h})\}_{(k,h)\in[K]\times[H]}$, where $r_{k,h}$ is sampled from some unknown reward distribution with expectation equal to $r_h^i(s_{k,h}, a_{k,h}, b_{k,h})$. Here, r^i denotes the unknown reward function of the i^{th} task. The goal is to compute nearly-optimal policies for N tasks under $\mathcal{M}$ simultaneously given the augmented datasets $\{\mathcal{D}^i\}_{i\in[N]}$.

There are strong practical motivations for considering the reward-free setting.

- In applications such as robotics, we face multiple tasks in sequential systems with shared transition dynamics (i.e. the world) but very different rewards. There, we prefer to learn the underlying transition independent of reward information.

Algorithm 2 Optimistic Value Iteration with Zero Reward (VI-Zero)

```
Require: Bonus $\beta_t$.
1: Initialize: for any $(s,a,b,h)$, $\widetilde{V}_h(s,a,b) \leftarrow H$, $\Delta \leftarrow H$, $N_h(s,a,b) \leftarrow 0$.
2: for episode $k = 1, \ldots, K$ do
3:    for step $h = H, H-1, \ldots, 1$ do
4:       for $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ do
5:          $t \leftarrow N_h(s,a,b)$.
6:          if $t > 0$ then
7:             $\widetilde{Q}_h(s,a,b) \leftarrow \min\{(\widehat{\mathbb{P}}_h \widetilde{V}_{h+1})(s,a,b) + \beta_t, H\}$.
8:       for $s \in \mathcal{S}$ do
9:          $\pi_h(s) \leftarrow \arg\max_{(a,b)\in\mathcal{A}\times\mathcal{B}} \widetilde{Q}_h(s,a,b)$.
10:         $\widetilde{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h} \widetilde{Q}_h)(s)$.
11:   if $\widetilde{V}_1(s_1) < \Delta$ then
12:      $\Delta \leftarrow \widetilde{V}_1(s_1)$ and $\widehat{\mathbb{P}}^{\mathrm{out}} \leftarrow \widehat{\mathbb{P}}$.
13:   for step $h = 1, \ldots, H$ do
14:      Take action $(a_h, b_h) \sim \pi_h(\cdot, \cdot | s_h)$, observe next state $s_{h+1}$.
15:      Add 1 to $N_h(s_h, a_h, b_h)$ and $N_h(s_h, a_h, b_h, s_{h+1})$.
16:      $\widehat{\mathbb{P}}_h(\cdot | s_h, a_h, b_h) \leftarrow N_h(s_h, a_h, b_h, \cdot)/N_h(s_h, a_h, b_h)$.
17: Output $\widehat{\mathbb{P}}^{\mathrm{out}}$.
```

- From the algorithm design perspective, decoupling exploration and planning (i.e. performing exploration without reward information) can be valuable for designing new algorithms in more challenging settings (e.g., with function approximation).

3.3.1 Algorithm description

We now describe our algorithm for reward-free learning in zero-sum Markov games.

Exploration phase. In the first phase of reward-free learning, we deploy algorithm Optimistic Value Iteration with Zero Reward (VI-Zero, Algorithm 2). This algorithm differs from the reward-aware Nash-VI (Algorithm 1) in two important aspects. First, we use zero reward in the exploration phase (Line 7), and only maintains an upper bound of the (reward-free) value function instead of both upper and lower bounds. Second, our exploration policy is the maximizing (instead of CCE) policy of the value function (Line 9). We remark that the $\widetilde{Q}_h(s,a,b)$ maintained in the algorithm 2 is no longer an upper bound for any actual value function (as it has no reward), but rather a measure of uncertainty or suboptimality that the agent may suffer—if she takes action (a, b) at state s and step h, and makes decisions by utilizing the empirical estimate $\widehat{\mathbb{P}}$ in the remaining steps (see a rigorous version of this statement in Lemma 75). Finally, the empirical transition $\widehat{\mathbb{P}}$ of the episode that minimizes $\widetilde{V}_1(s_1)$ is outputted and passed to the planning phase.

Planning phase. After obtaining the estimate of tranisiton $\widehat{\mathbb{P}}$, our planning algorithm is rather simple. For the i^{th} task, let $\widehat{r}^i$ be the empirical estimate of r^i computed using the i^{th} augmented dataset $\mathcal{D}^i$. Then we compute the Nash equilibrium of the Markov game $\mathcal{M}(\widehat{\mathbb{P}}, \widehat{r}^i)$ with estimated transition $\widehat{\mathbb{P}}$ and reward $\widehat{r}^i$. Since both $\widehat{\mathbb{P}}$ and $\widehat{r}^i$ are known exactly, this is a pure computation problem without any sampling error and can be efficiently solved by simple planning algorithms such as the vanilla Nash value iteration without optimism (see Appendix B.2.2 for more details).

3.3.2 Theoretical guarantees

Now we are ready to state our theoretical guarantees for reward-free learning. It claims that the empirical transition $\widehat{\mathbb{P}}^{\text{out}}$ outputted by VI-Zero is close to the true transition $\mathbb{P}$, in the sense that any Nash equilibrium of the $\mathcal{M}(\widehat{\mathbb{P}}, \widehat{r}^i)$ $(i \in [N])$ is also an approximate Nash equilibrium of the true underlying Markov game $\mathcal{M}(\mathbb{P}, r^i)$, where $\widehat{r}^i$ is the empirical estimate of r^i computed using $\mathcal{D}^i$.

Theorem 18 (Sample complexity of VI-Zero)**.** *There exists an absolute constant c, for any $p \in (0, 1]$, $\varepsilon \in (0, H]$, $N \in \mathbb{N}$, if we choose bonus $\beta_t = c(\sqrt{H^2\iota/t} + H^2S\iota/t)$ with $\iota = \log(NSABT/p)$ and $K \geq c(H^4SAB\iota/\varepsilon^2 + H^3S^2AB\iota^2/\varepsilon)$, then with probability at least $1 - p$, the output $\widehat{\mathbb{P}}^{\text{out}}$ of Algorithm 2 satisfies: For any N fixed reward functions $r^1, \ldots, r^N$, a Nash equilibrium of Markov game $\mathcal{M}(\widehat{\mathbb{P}}^{\text{out}}, \widehat{r}^i)$ is also an ε-approximate Nash equilibrium of the true Markov game $\mathcal{M}(\mathbb{P}, r^i)$ for all $i \in [N]$.*

Theorem 18 shows that, when ε is small, VI-Zero only needs $\tilde{\mathcal{O}}(H^4SAB/\varepsilon^2)$ samples to learn an estimate of the transition $\widehat{\mathbb{P}}^{\text{out}}$, which is accurate enough to learn the approximate Nash equilibrium for any N fixed rewards. The most important advantage of reward-free learning comes from the sample complexity only scaling polylogarithmically with respect to the number of tasks or reward functions N. This is in sharp contrast to the reward-aware algorithms (e.g. Nash-VI), where the algorithm has to be rerun for each different task, and the total sample complexity must scale linearly in N. In exchange for this benefit, compared to Nash-VI, VI-Zero loses a factor of H in the leading term of sample complexity since we cannot use Bernstein bonus anymore due to the lack of reward information. VI-Zero also does not have a regret guarantee, since again without reward information, the exploration policies are naturally sub-optimal. The proof of Theorem 18 can be found in Appendix B.2.1.

Connections with reward-free learning in MDPs. Since MDPs are special cases of Markov games, our algorithm VI-Zero directly applies to the single-agent setting, and yields a sample complexity similar to existing results [Zhang et al., 2020b, Wang et al., 2020].

However, distinct from existing results which require both the exploration algorithm and the planning algorithm to be specially designed to work together, our algorithm allows an arbitrary planning algorithm as long as it computes the Nash equilibrium of a Markov game with *known* transition and reward. Therefore, our results completely decouple the exploration and the planning.

Lower bound for reward-free learning. Finally, we comment that despite the sample complexity in Theorem 18 scaling as AB instead of $A + B$, our next theorem states that unlike the general reward-aware setting, this AB scaling is unavoidable in the reward-free setting. This reveals an intrinsic gap between the reward-free and reward-aware learning: An $A+B$ dependency is only achievable via sampling schemes that are reward-aware. A similar lower bound is also presented in Zhang et al. [2020a] for the discounted setting with a different hard instance construction.

Theorem 19 (Lower bound for reward-free learning of Markov games)**.** *There exists an absolute constant $c > 0$ such that for any $\varepsilon \in (0, c]$, there exists a family of Markov games $\mathfrak{M}(\varepsilon)$ satisfying that: for any reward-free algorithm $\mathfrak{A}$ using $K \le cH^2SAB/\varepsilon^2$*

episodes, there exists a Markov game $\mathcal{M} \in \mathfrak{M}(\varepsilon)$ such that if we run $\mathfrak{A}$ on $\mathcal{M}$ and output policies $(\widehat{\mu}, \widehat{\nu})$, then with probability at least $1/4$, we have $(V_1^{\dagger,\widehat{\nu}} - V_1^{\widehat{\mu},\dagger})(s_1) \geq \varepsilon$.

This lower bound shows that the sample complexity in Theorem 18 is optimal in S, A, B, and ε. The proof of Theorem 19 can be found in Appendix B.2.3.

3.4 Multi-player general-sum games

In this section, we generalize the algorithms and their theoretical guarantees to general games.

3.4.1 Multiplayer optimistic Nash value iteration

Here we present the Multi-Nash-VI algorithm, which is an extension of Algorithm 1 for multi-player general-sum Markov games.

The Equilibrium subroutine. Our Equilibrium subroutine in Line 11 could be taken from either one of the $\{$Nash, CE, CCE$\}$ subroutines for *one-step* games. When using Nash, we compute the Nash equilibrium of a one-step multi-player game (see, e.g., Berg and Sandholm [2016] for an overview of the available algorithms); the worst-case computational complexity of such a subroutine will be PPAD-hard [Daskalakis, 2013]. When using CE or CCE, we find CEs or CCEs of the one-step games respectively, which can be solved in polynomial time using linear programming. However, the policies found are not guaranteed to be a product policy. We remark that in Algorithm 1 we used the CCE subroutine for finding Nash in two-player zero-sum games, which seemingly contrasts the principle of using the right subroutine for finding the right equilibrium, but nevertheless works as the Nash equilibrium and CCE are equivalent in zero-sum games.

Now we are ready to present the theoretical guarantees for Algorithm 3. We let π^k denote the policy computed in line 11 of Algorithm 3 in the k^{th} episode.

Algorithm 3 Multiplayer Optimistic Nash Value Iteration (Multi-Nash-VI)

```
Initialize: for any $(s, \boldsymbol{a}, h, i)$, $\overline{Q}_{h,i}(s, \boldsymbol{a}) \leftarrow H$, $\underline{Q}_{h,i}(s, \boldsymbol{a}) \leftarrow 0$, $\Delta \leftarrow H$, $N_h(s, \boldsymbol{a}) \leftarrow 0$.
for episode $k = 1, \ldots, K$ do
    for step $h = H, H-1, \ldots, 1$ do
        for $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ do
            $t \leftarrow N_h(s, \boldsymbol{a})$;
            if $t > 0$ then
                for player $i = 1, 2, \ldots, m$ do
                    $\overline{Q}_{h,i}(s, \boldsymbol{a}) \leftarrow \min\{(r_{h,i} + \widehat{\mathbb{P}}_h \overline{V}_{h+1,i})(s, \boldsymbol{a}) + \beta_t, H\}$.
                    $\underline{Q}_{h,i}(s, \boldsymbol{a}) \leftarrow \max\{(r_{h,i} + \widehat{\mathbb{P}}_h \underline{V}_{h+1,i})(s, \boldsymbol{a}) - \beta_t, 0\}$.
        for $s \in \mathcal{S}$ do
            $\pi_h(\cdot|s) \leftarrow \textsc{Equilibrium}(\overline{Q}_{h,1}(s, \cdot), \overline{Q}_{h,2}(s, \cdot), \cdots, \overline{Q}_{h,M}(s, \cdot))$.
            for player $i = 1, 2, \ldots, m$ do
                $\overline{V}_{h,i}(s) \leftarrow (\mathbb{D}_{\pi_h} \overline{Q}_{h,i})(s)$;   $\underline{V}_{h,i}(s) \leftarrow (\mathbb{D}_{\pi_h} \underline{Q}_{h,i})(s)$.
    if $\max_{i \in [m]}(\overline{V}_{1,i} - \underline{V}_{1,i})(s_1) < \Delta$ then
        $\Delta \leftarrow \max_{i \in [m]}(\overline{V}_{1,i} - \underline{V}_{1,i})(s_1)$ and $\pi^{\mathrm{out}} \leftarrow \pi$.
    for step $h = 1, \ldots, H$ do
        Take action $\boldsymbol{a}_h \sim \pi_h(\cdot|s_h)$, observe reward $r_h$ and next state $s_{h+1}$.
        Add 1 to $N_h(s_h, \boldsymbol{a}_h)$ and $N_h(s_h, \boldsymbol{a}_h, s_{h+1})$.
        $\widehat{\mathbb{P}}_h(\cdot|s_h, \boldsymbol{a}_h) \leftarrow N_h(s_h, \boldsymbol{a}_h, \cdot)/N_h(s_h, \boldsymbol{a}_h)$.
Output $\pi^{\mathrm{out}}$.
```

Theorem 20 (Multi-Nash-VI)**.** *There exists an absolute constant c, for any $p \in (0, 1]$, let $\iota = \log(SABT/p)$, then with probability at least $1 - p$, Algorithm 3 with bonus $\beta_t = c\sqrt{SH^2\iota/t}$ and* Equilibrium *being one of* $\{\textsc{Nash}, \mathrm{CE}, \mathrm{CCE}\}$ *satisfies (repsectively):*

- *π^{out} is an ε-approximate $\{$*Nash,CE,CCE*$\}$, if the number of episodes $K \geq \Omega(H^4 S^2 (\prod_{i=1}^m A_i)\iota/\varepsilon^2)$.*

- $\mathfrak{R}_{\{\mathsf{Nash,CE,CCE}\}}(K) \leq \mathcal{O}(\sqrt{H^3 S^2 (\prod_{i=1}^m A_i) T\iota})$.

In the situation where the Equilibrium subroutine is taken as Nash, Theorem 20 provides the sample complexity bound of Multi-Nash-VI algorithm to find an ε-approximate Nash equilibrium and its regret bound. Compared with our earlier result in two-player zero-sum games (Theorem 16), here the sample complexity scales as S^2H^4 instead of SH^3. This is because the auxiliary bonus and Bernstein concentra-

Algorithm 4 Multiplayer Optimistic Value Iteration with Zero Reward (Multi-VI-Zero)

```
**Initialize:** for any $(s, \boldsymbol{a}, h)$, $\widetilde{V}_h(s, \boldsymbol{a}) \leftarrow H$, $\Delta \leftarrow H$, $N_h(s, \boldsymbol{a}) \leftarrow 0$.
**for** episode $k = 1, \ldots, K$ **do**
    **for** step $h = H, H-1, \ldots, 1$ **do**
        **for** $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ **do**
            $t \leftarrow N_h(s, \boldsymbol{a})$.
            **if** $t > 0$ **then**
                $\widetilde{Q}_h(s, \boldsymbol{a}) \leftarrow \min\{(\widehat{\mathbb{P}}_h \widetilde{V}_{h+1})(s, \boldsymbol{a}) + \beta_t, H\}$.
        **for** $s \in \mathcal{S}$ **do**
            $\pi_h(s) \leftarrow \arg\max_{\boldsymbol{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_m} \widetilde{Q}_h(s, \boldsymbol{a})$.
            $\widetilde{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h} \widetilde{Q}_h)(s)$.
    **if** $\widetilde{V}_1(s_1) < \Delta$ **then**
        $\Delta \leftarrow \widetilde{V}_1(s_1)$ and $\widehat{\mathbb{P}}^{\mathrm{out}} \leftarrow \widehat{\mathbb{P}}$.
    **for** step $h = 1, \ldots, H$ **do**
        take action $\boldsymbol{a}_h \sim \pi_h(\cdot, \cdot | s_h)$, observe next state $s_{h+1}$.
        add 1 to $N_h(s_h, \boldsymbol{a}_h)$ and $N_h(s_h, \boldsymbol{a}_h, s_{h+1})$.
        $\widehat{\mathbb{P}}_h(\cdot | s_h, \boldsymbol{a}_h) \leftarrow N_h(s_h, \boldsymbol{a}_h, \cdot) / N_h(s_h, \boldsymbol{a}_h)$.
**Output** $\widehat{\mathbb{P}}^{\mathrm{out}}$.
```

tion technique do not apply here. Furthermore, the sample complexity is proportional to $\prod_{i=1}^{m} A_i$, which increases exponentially as the number of players increases.

Runtime of Algorithm 3 We remark that while the Nash guarantee is the strongest among the three guarantees presented in Theorem 20, the runtime of Algorithm 3 in the Nash case is not guaranteed to be polynomial and in the worst case PPAD-hard (due to the hardness of the NASH subroutine). In contrast, the CE and CCE guarantees are weaker, but the corresponding algorithms are guaranteed to finish in polynomial time.

3.4.2 Multiplayer reward-free learning

We can also generalize VI-Zero to the multiplayer setting and obtain Algorithm 4, Multi-VI-Zero, which is almost the same as VI-Zero except that its exploration bonus β_t is larger than that of VI-Zero by a $\sqrt{S}$ factor.

Similar to Theorem 18, we have the following theoretical guarantee claiming that any {Nash,CCE,CE} of the $\mathcal{M}(\widehat{\mathbb{P}}, \widehat{r}^i)$ $(i \in [N])$ is also an approximate {Nash,CCE,CE} of the true Markov game $\mathcal{M}(\mathbb{P}, r^i)$, where $\widehat{\mathbb{P}}^{\mathrm{out}}$ is the empirical transition outputted by Algorithm 4 and $\widehat{r}^i$ is the empirical estimate of r^i.

Theorem 21 (Multi-VI-Zero). *There exists an absolute constant c, for any $p \in (0,1]$, $\varepsilon \in (0, H]$, $N \in \mathbb{N}$, if we choose bonus $\beta_t = c\sqrt{H^2 S\iota/t}$ with $\iota = \log(NSABT/p)$ and $K \geq c(H^4 S^2 (\prod_{i=1}^m A_i)\iota/\varepsilon^2)$, then with probability at least $1-p$, the output $\widehat{\mathbb{P}}^{out}$ of Algorithm 4 has the following property: for any N fixed reward functions $r^1, \ldots, r^N$, any {*Nash*,CCE,CE*} of Markov game $\mathcal{M}(\widehat{\mathbb{P}}^{out}, \widehat{r}^i)$ is also an ε-approximate {*Nash*,CCE,CE*} of the true Markov game $\mathcal{M}(\mathbb{P}, r^i)$ for all $i \in [N]$.*

The proof of Theorem 21 can be found in Appendix B.3.2. It is worth mentioning that the empirical Markov game $\mathcal{M}(\widehat{\mathbb{P}}^{\mathrm{out}}, \widehat{r}^i)$ may have multiple {Nash equilibria,CCEs,CEs} and Theorem 21 ensures that all of them are ε-approximate {Nash equilibria,CCEs,CEs} of the true Markov game. Also, note that the sample complexity here is quadratic in the number of states because we are using the exploration bonus $\beta_t = \sqrt{H^2 S\iota/t}$ that is larger than usual by a $\sqrt{S}$ factor.

Chapter 4

Markov Games: Model-free Learning

Chapter 3 introduces a sample-efficient method for learning in Markov games using a model-based approach. While this approach offers promise, it also presents several challenges that have to be addressed.

One such challenge is *the curse of multiagents*—let A_i be the number of actions for the i-th player, then the number of possible joint actions (as well as the number of parameters to specify a Markov game) scales with $\prod_{i=1}^{m} A_i$, which grows exponentially with the number of agents m. This remains to be a bottleneck even for the best existing algorithms for learning Markov games. In fact, a majority of these algorithms adapt the classical single-agent algorithms, such as value iteration or Q-learning, into the multiagent setting Bai et al. [2020], Liu et al. [2021], whose sample complexity scales at least linearly with respect to $\prod_{i=1}^{m} A_i$. This is prohibitively large in practice even for fairly small multiagent applications, say only ten agents are involved with ten actions available for each agent.

Another remaining challenge is to design *decentralized* algorithms. While a centralized algorithm requires the existence of a centralized controller which gathers all information and jointly optimizes the policies of all agents, a decentralized algorithm allows each agent to only observe her own actions and rewards while optimizing her own policy independently. Decentralized algorithms are usually preferred over centralized algorithms in practice since

1. Decentralized algorithms are typically easier to implement as we only need to implement single-agent algorithms for each player without complex interactions;

2. Decentralized algorithms are more versatile as the individual learners are indifferent to the interaction and the number of other agents; and

3. They are also faster in the systems where communication is the bottleneck, due to less communication required.

While several provable decentralized MARL algorithms have been developed [see, e.g., Zhang et al., 2018, Sayin et al., 2021, Daskalakis et al., 2020], they either have only asymptotic guarantees or work only under certain reachability assumptions (see Section 1.2). The existing provably *efficient* algorithms for general Markov games (without further assumptions) are exclusively centralized algorithms [Bai and Jin, 2020, Xie et al., 2020, Liu et al., 2021].

This motivates us to ask the following open question:

Can we design *decentralized* MARL algorithms that *break the curse of multiagents*?

This chapter addresses both challenges mentioned above, and provides the first positive answer to this question in the basic setting of tabular episodic Markov games. We propose a new class of single-agent RL algorithms—V-learning, which converts any adversarial bandit algorithm with suitable regret guarantees into an RL algorithm. Similar to the classical Q-learning algorithm, V-learning also performs incremental updates to the values. Different from Q-learning, V-learning only maintains the V-value functions instead of the Q-value functions. We remark that the number of parameters of Q-value functions in MARL is $\mathcal{O}(S\prod_{i=1}^{m} A_i)$, where S is the number of states, while the number of parameters of V-value functions is only $\mathcal{O}(S)$. This key difference allows V-learning to be readily extended to the MARL setting by simply letting all agents run V-learning independently, which gives a fully *decentralized* algorithm.

Algorithm 5 V-LEARNING

```
1: Initialize: for any $(s, a, h)$, $V_h(s) \leftarrow H + 1 - h$, $N_h(s) \leftarrow 0$, $\pi_h(a|s) \leftarrow 1/A$.
2: for episode $k = 1, \ldots, K$ do
3:     receive $s_1$.
4:     for step $h = 1, \ldots, H$ do
5:         take action $a_h \sim \pi_h(\cdot|s_h)$, observe reward $r_h$ and next state $s_{h+1}$.
6:         $t = N_h(s_h) \leftarrow N_h(s_h) + 1$.
7:         $\tilde{V}_h(s_h) \leftarrow (1 - \alpha_t)\tilde{V}_h(s_h) + \alpha_t(r_h + V_{h+1}(s_{h+1}) + \beta_t)$.
8:         $V_h(s_h) \leftarrow \min\{H + 1 - h, \tilde{V}_h(s_h)\}$.
9:         $\pi_h(\cdot|s_h) \leftarrow$ ADV_BANDIT_UPDATE$(a_h, \frac{H - r_h - V_{h+1}(s_{h+1})}{H})$ on $(s_h, h)$-th adversarial bandit.
```

4.1 V-Learning Algorithm

In this section, we introduce the V-learning algorithm as a new class of single-agent RL algorithms, which converts any adversarial bandit algorithm with suitable regret guarantees into an RL algorithm. We also present its theoretical guarantees for finding a nearly optimal policy in the single-agent setting.

4.1.1 Training algorithm

To begin with, we describe the V-learning algorithm (Algorithm 5). It maintains a value $V_h(s)$, a counter $N_h(s)$, and a policy $\pi_h(\cdot|s)$ for each state s and step h, and initializes them to be the max value, 0, and uniform distribution respectively. V-learning also instantiates $S \times H$ different adversarial bandit algorithms—one for each (s, h) pair. At each step h in each episode k, the algorithm performs three major steps:

- Policy execution (Line 5-6): the algorithm takes action a_h according to the maintained π_h, then observes the reward r_h and the next state s_{h+1}, and increases the counter $N_h(s_h)$ by 1.

- V-value update (Line 7-8): the algorithm performs incremental update to the value function:

$$\tilde{V}_h(s_h) \leftarrow (1 - \alpha_t)\tilde{V}_h(s_h) + \alpha_t(r_h + V_{h+1}(s_{h+1}) + \beta_t) \tag{4.1}$$

Protocol 6 ADVERSARIAL BANDIT ALGORITHM

```
Initialize: for any $b$, $\theta_1(b) \leftarrow 1/B$.
for step $t = 1, \ldots, T$ do
    adversary chooses loss $\ell_t$.
    take action $b_t \sim \theta_t$, observe noisy bandit-feedback $\tilde{\ell}_t(b_t)$.
    $\theta_{t+1} \leftarrow$ ADV_BANDIT_UPDATE$(b_t, \tilde{\ell}_t(b_t))$.
```

where α_t is the learning rate, and β_t is the bonus to promote optimism (and exploration). The choices of both quantities will be specified later. Next, we simply update V_h as a truncated version of $\tilde{V}_h$.

- Policy update (Line 9): the algorithm feeds the action a_h and its "loss" $\frac{H-r_h-V_{h+1}(s_{h+1})}{H}$ to the (s_h, h)-th adversarial bandit algorithm, and receives the updated policy $\pi_h(\cdot|s_h)$.

Throughout this paper, we will always use the following learning rate α_t. We also define an auxiliary sequence $\{\alpha_t^i\}_{i=1}^t$ based on the learning rate, which will be frequently used across the paper.

$$\alpha_t = \frac{H+1}{H+t}, \quad \alpha_t^0 = \prod_{j=1}^{t}(1-\alpha_j), \quad \alpha_t^i = \alpha_i \prod_{j=i+1}^{t}(1-\alpha_j). \tag{4.2}$$

We remark that our incremental update (4.1) bears significant similarity to Q-learning, and our choice of learning rate is precisely the same as the choice in Q-learning [Jin et al., 2018]. However, a key difference is that the V-learning algorithm maintains V-value functions instead of Q-value functions. This is crucial when extending V-learning to the multiplayer setting where the number of parameters of Q-value functions becomes $\mathcal{O}(HS\prod_{i=1}^m A_i)$ while the number of parameters of V-value functions is only $\mathcal{O}(HS)$. Since V-learning does not use action-value functions, it resorts to adversarial bandit algorithms to update its policy.

ADV_BANDIT_UPDATE subroutine: Consider a multi-arm bandit problem with adversarial loss, where we denote the action set by $\mathcal{B}$ with $|\mathcal{B}| = B$. At round t, the learner picks a strategy (distribution over actions) $\theta_t \in \Delta_{\mathcal{B}}$, and the adversary

Algorithm 7 Executing Output Policy $\widehat{\pi}$ of V-learning

```
sample $k \leftarrow \mathrm{Uniform}([K])$.
for step $h = 1, \ldots, H$ do
    observe $s_h$, and set $t \leftarrow N_h^k(s_h)$.
    set $k \leftarrow k_h^i(s_h)$, where $i \in [t]$ is sampled with probability $\alpha_t^i$.
    take action $a_h \sim \pi_h^k(\cdot|s_h)$.
```

chooses a loss vector $\ell_t \in [0,1]^B$. Then the learner takes an action b_t that is sampled from distribution θ_t, and receives a noisy bandit-feedback $\tilde{\ell}_t(b_t) \in [0,1]$ where $\mathbb{E}[\tilde{\ell}_t(b_t)|\ell_t, b_t] = \ell_t(b_t)$. Then, the adversarial bandit algorithm performs updates based on b_t and $\tilde{\ell}_t(b_t)$, and outputs the strategy for next round θ_{t+1}, which we abstract as $\theta_{t+1} \leftarrow \textsc{Adv\_Bandit\_Update}(b_t, \tilde{\ell}_t(b_t))$ (see Protocol 6).

4.1.2 Output policy

We define the final output policy $\widehat{\pi}$ of V-learning by how to execute this policy (see Algorithm 7). Let V^k, N^k, π^k be the value, counter and policy maintained by the V-learning algorithm at *the beginning* of episode k. The output policy maintains a scalar k, which is initially uniformly sampled from $[K]$. At each step h, after observing s_h, $\widehat{\pi}$ plays a mixture of policy $\{\pi_h^{k^i}(\cdot|s_h)\}_{i=1}^t$ with corresponding probability $\{\alpha_t^i\}_{i=1}^t$ defined in (4.2). Here $t = N_h^k(s_h)$ is the number of times s_h is visited at step h at the beginning of episode k, and k^i is short for $k_h^i(s_h)$ which is the index of the episode when s_h is visited at step h for the i-th time. After that, $\widehat{\pi}$ sets k to be the index $k_h^i(s_h)$ whose policy is just played within the mixture, and continues the same process for the next step. This mixture form of output policy $\widehat{\pi}$ is mainly due to the incremental updates of V-learning. One can show that, if omitting the optimistic bonus, $V_1^K(s_1)$ computed in the V-learning algorithm is a stochastic estimate of the value of policy $\widehat{\pi}$. See Figure 4-1 for a comparison between the certified policy techinique and conventional online-to-batch conversion.

We remark that $\widehat{\pi}$ is not a Markov policy, but a general random policy (see Definition in Section 2.1.1), which can be written as a set of maps $\{\pi_h : \Omega \times \mathcal{S}^h \to \mathcal{A}_i\}$. The choice of action at each step h depends on a joint randomness $\omega \in \Omega$ which is

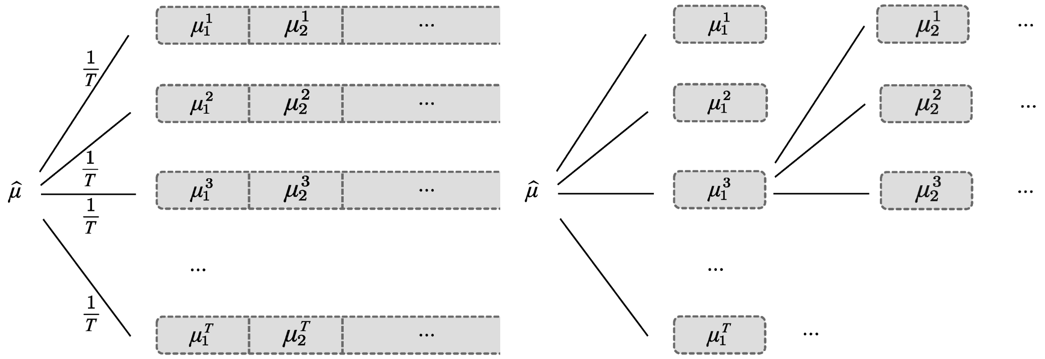


Figure 4-1: A comparison between the conventional online-to-batch conversion (left) and the certified policy (right). While conventional online-to-batch conversion uses uniform weighting and episode-wise average, certified policy uses non-uniform reweighting and step-wise average.

shared among all steps, and the history of past states $(s_1, \ldots, s_h)$. In Section 4.4, we will further introduce a simple monotone technique that allows V-learning to output a Markov policy in both the single-agent and the two-player zero-sum setting.

4.1.3 Single-agent guarantees

We first state our requirement for the adversarial bandit algorithm used in V-learning, which is to have a high probability *weighted* external regret guarantee as follows. The weights $\{\alpha_t^i\}_{i=1}^t$ are defined in (4.2).

Assumption 1. For any $t \in \mathbb{N}$ and any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$\max_{\theta \in \Delta_{\mathcal{B}}} \sum_{i=1}^{t} \alpha_t^i [\langle \theta_i, \ell_i \rangle - \langle \theta, \ell_i \rangle] \leq \xi(B, t, \log(1/\delta)). \tag{4.3}$$

We further assume the existence of an upper bound $\Xi(B, t, \log(1/\delta)) \geq \sum_{t'=1}^{t} \xi(B, t', \log(1/\delta))$ where

- $\xi(B, t, \log(1/\delta))$ is non-decreasing in B for any t, δ;

- $\Xi(B, t, \log(1/\delta))$ is concave in t for any B, δ.

Assumption 1 can be satisfied by modifying many existing algorithms with unweighted external regret to the weighted setting. In particular, we prove that the Follow-the-Regularized-Leader (FTRL) algorithm (Algorithm 19) satisfies the Assumption 1 with bounds

$$\xi(B,t,\log(1/\delta)) \le \mathcal{O}(\sqrt{HB\log(B/\delta)/t}), \quad \Xi(B,t,\log(1/\delta)) \quad \le \mathcal{O}(\sqrt{HBt\log(B/\delta)}).$$

The H factor comes into the bounds because our choice of weights $\{\alpha_t^i\}$ in (4.2) involves H. We refer readers to Appendix C.6 for more details.

We are now ready to introduce the theoretical guarantees of V-learning for finding near-optimal policies in the single-agent setting.

Theorem 22. *Suppose subroutine* ADV_BANDIT_UPDATE *satisfies Assumption 1. For any $\delta \in (0,1)$ and $K \in \mathbb{N}$, let $\iota = \log(HSAK/\delta)$. Choose learning rate α_t according to (4.2) and bonus $\{\beta_t\}_{t=1}^K$ so that $\sum_{i=1}^t \alpha_t^i \beta_i = \Theta(H\xi(A,t,\iota) + \sqrt{H^3\iota/t})$ for any $t \in [K]$. Then, with probability at least $1-\delta$, after running Algorithm 5 for K episodes, the output policy $\widehat{\pi}$ of Algorithm 7 satisfies*

$$V_1^\star(s_1) - V_1^{\widehat{\pi}}(s_1) \le \mathcal{O}((H^2S/K)\cdot \Xi(A,K/S,\iota) + \sqrt{H^5S\iota/K}).$$

In particular, when instantiating subroutine ADV_BANDIT_UPDATE *by FTRL (Algorithm 19), we can choose $\beta_t = c\cdot\sqrt{H^3A\iota/t}$ for some absolute constant c, where $V_1^\star(s_1) - V_1^{\widehat{\pi}}(s_1) \le \mathcal{O}(\sqrt{H^5SA\iota/K})$.*

The special cases of Theorem 22 and Theorem 23 (when the subroutine is instantiated by FTRL) were firstly presented [Bai et al., 2020], with an additional $\sqrt{H}$ factor in the error due to a looser choice of hyperparameter.

Theorem 22 characterizes how fast the suboptimality of $\widehat{\pi}$ decreases with respect to the total number of episodes K. In particular, to obtain an ε-optimal output policy $\widehat{\pi}$, we only need to use a number of episodes $K = \tilde{\mathcal{O}}(H^5SA/\varepsilon^2)$. This is H^2 factor larger than the information-theoretic lower bound $\Omega(H^3SA/\varepsilon^2)$ in this setting [Jin et al., 2018]. We remark that one extra H factor is due to the incremental update and the use of learning rate in (4.2) which is exactly the same for Q-learning algorithm [Jin et al., 2018]. The other H factor can be potentially improved by using refined first-order regret bound in V-learning. A first-order regret bound Agarwal et al. [2017] is a data-dependent regret bound that depends on the minimum total loss incurred instead of the total number of steps. This makes it possible to further combine the current analysis with Bernstein-type concentration and a sharper total variance bound Azar et al. [2017]. We leave this as future work.

While V-learning seems to be no better than classical value iteration or Q-learning in the single-agent setting, its true power starts to show up in the multiagent setting: Value iteration and Q-learning require highly nontrivial efforts to adapt them to the multiagent setting, and by design they suffer from the curse of multiagents [Bai et al., 2020, Liu et al., 2021]. In the following sections, we will show that V-learning can be directly extended to the multiagent setting by simply letting all agents run V-learning independently. Furthermore, V-learning breaks the curse of multiagents.

4.2 Two-player Zero-sum Markov Games

In this section, we provide the sample efficiency guarantee for V-learning to find Nash equilibria in two-player zero-sum Markov games.

4.2.1 Finding Nash equilibria

In the two-player zero-sum setting, we have two agents whose rewards satisfy $r_{1,h} = -r_{2,h}$ for any $h \in [H]$. Our algorithm is simply that both agents run V-learning (Algorithm 5) independently with learning rate α_t as specified in (4.2). Each player j will uses her own set of bonus $\{\beta_{j,t}\}$ that depends on the number of her actions and will be specified later. To execute the output policy, both agents simply execute Algorithm 7 independently using their own intermediate policies computed by V-learning.

We have the following theorem for V-learning. For clean presentation, we denote $A = \max_{j\in[2]} A_j$.

Theorem 23. *Suppose subroutine* Adv_Bandit_Update *satisfies Assumption 1. For any* $\delta \in (0, 1)$ *and* $K \in \mathbb{N}$, *let* $\iota = \log(HSAK/\delta)$. *Choose learning rate* α_t *according to* (4.2) *and bonus* $\{\beta_{j,t}\}_{t=1}^K$ *of the* j*-th player so that* $\sum_{i=1}^t \alpha_t^i \beta_{j,i} = \Theta(H\xi(A_j, t, \iota) + \sqrt{H^3\iota/t})$ *for any* $t \in [K]$. *After running Algorithm 5 for* K *episodes, let* $\widehat{\pi}_1, \widehat{\pi}_2$ *be the output policies by Algorithm 7 for each player. Then with probability at least* $1 - \delta$, *the product policy* $\widehat{\pi} = \widehat{\pi}_1 \times \widehat{\pi}_2$ *satisfies*

$$\max_{j\in[2]}[V_{j,1}^{\dagger,\widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \le \mathcal{O}((H^2S/K) \cdot \Xi(A, K/S, \iota) + \sqrt{H^5S\iota/K}).$$

When instantiating Adv_Bandit_Update *by FTRL (Algorithm 19), we can choose* $\beta_{j,t} = c \cdot \sqrt{H^3A_j\iota/t}$ *for some absolute constant c, which leads to*

$$\max_{j\in[2]}[V_{j,1}^{\dagger,\widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \le \mathcal{O}(\sqrt{H^5SA\iota/K}).$$

Theorem 23 claims that, to find an $\varepsilon-$approximate Nash equilibrium, we only need to use a number of episodes $K = \tilde{\mathcal{O}}(H^5SA/\varepsilon^2)$, where $A = \max_{j\in[2]} A_j$. In contrast, value iteration or Q-learning-based algorithms require at least $\Omega(H^3SA_1A_2/\varepsilon^2)$ episodes to find Nash equilibria Bai et al. [2020], Liu et al. [2021]. Furthermore, V-learning is a fully decentralized algorithm. To our best knowledge, V-learning is the only algorithm up to today that achieves sample complexity linear in A for finding Nash equilibrium in two-player zero-sum Markov games.

We remark that V-learning only performs $\mathcal{O}(1)$ operations and calls subroutine Adv_Bandit_Update once every time a new sample is observed. As long as the adversarial bandit algorithm used in V-learning is computationally efficient (which is the case for FTRL), V-learning itself is also computationally efficient.

4.3 Multiplayer General-sum Markov Games

In multiplayer general-sum games, finding Nash equilibria is computationally hard in general (which is technically PPAD-complete Daskalakis [2013]). In this section, we focus on finding two commonly-used alternative notions of equilibria in the game theory—coarse correlated equilibria and correlated equilibria. Both are relaxed notions of Nash equilibria.

4.3.1 Finding coarse correlated equilibria

The algorithm for finding CCE is again running V-learning (Algorithm 5) independently for each agent j with learning rate α_t (as specified in (4.2)) and bonus $\{\beta_{j,t}\}$ (to be specified later). The major difference from the case of finding Nash equilibria is that CCE and CE require the output policy to be a joint correlated policy. We achieve this correlation by feeding the same random seed to all agents at the very beginning when they execute the output policy according to Algorithm 7. That is, while training can be done in a fully decentralized fashion, we require one round of communication at the beginning of the execution to broadcast the shared random seed. After that, each agent can simply execute her own output policy independently. During the execution, since the states visited are shared among all agents, shared random seed allows the same index i to be sampled across all agents in the Step 4 of Algorithm 7 at every step. We denote this correlated joint output policy as $\widehat{\pi} = \widehat{\pi}_1 \odot \ldots \odot \widehat{\pi}_m$.

We remark that to specify a correlated policy in general, we need to specify the probability for taking all action combinations $(a_1, \ldots, a_m)$ for each (s, h). This requires at least $\Omega(HS \prod_{j=1}^{m} A_j)$ space, which grows exponentially with the number of agents m. The way V-learning specifies the joint policy only requires agents to store their own intermediate counters and policies computed during training. This only takes a total of $\mathcal{O}(HSK(\sum_{j=1}^{m} A_j))$ space, which scales only linearly with the number of agents. Our approach dramatically improves over the former approach in space complexity when the number of agents is large.

We now present the guarantees for V-learning to learn a CCE as follows. Let $A = \max_{j\in[m]} A_j$.

Theorem 24. *Suppose subroutine* Adv_Bandit_Update *satisfies Assumption 1.*

For any $\delta \in (0,1)$ and $K \in \mathbb{N}$, let $\iota = \log(mHSAK/\delta)$. Choose learning rate α_t according to (4.2) *and bonus $\{\beta_{j,t}\}_{t=1}^{K}$ of the j-th player so that $\sum_{i=1}^{t} \alpha_t^i \beta_{j,i} = \Theta(H\xi(A_j, t, \iota) + \sqrt{H^3\iota/t})$ for any $t \in [K]$. After all the players running Algorithm 5 for K episodes, let $\widehat{\pi}_j$ be the output policy by Algorithm 7 for the j-th player. Then with probability at least $1-\delta$, the joint policy $\widehat{\pi} = \widehat{\pi}_1 \odot \ldots \odot \widehat{\pi}_m$ satisfies*

$$\max_{j\in[m]}[V_{j,1}^{\dagger,\widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \leq \mathcal{O}((H^2S/K)\cdot \Xi(A, K/S, \iota) + \sqrt{H^5S\iota/K}).$$

When instantiating ADV_BANDIT_UPDATE *by FTRL (Algorithm 19), we can choose $\beta_{j,t} = c \cdot \sqrt{H^3A_j\iota/t}$ for some absolute constant c, which leads to*

$$\max_{j\in[m]}[V_{j,1}^{\dagger,\widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \leq \mathcal{O}(\sqrt{H^5SA\iota/K}).$$

Theorem 24 claims that, to find an $\varepsilon-$approximate CCE, V-learning only needs to use a number of episodes $K = \tilde{\mathcal{O}}(H^5SA/\varepsilon^2)$, where $A = \max_{j\in[m]} A_j$. This is in sharp contrast to the prior results for multiplayer general-sum Markov games, which use value-iteration-based algorithms, and require at least $\Omega(H^4S^2(\prod_{i=1}^{m} A_i)/\varepsilon^2)$ episodes Liu et al. [2021]. As a result, V-learning is the first algorithm that breaks the curse of multiagents for finding CCE in Markov games.

4.3.2 Finding correlated equilibria

The algorithm for finding CE is almost the same as the algorithm for finding CCE except that we now require a different ADV_BANDIT_UPDATE subroutine, which has the following high probability weighted *swap regret* guarantee.

Assumption 2. For any $t \in \mathbb{N}$ and any $\delta > 0$, with probability at least $1-\delta$, we have

$$\max_{\psi\in\Psi} \sum_{i=1}^{t} \alpha_t^i[\langle\theta_i, l_i\rangle - \langle\psi \diamond \theta_i, l_i\rangle] \leq \xi_{\rm sw}(B, t, \log(1/\delta)). \tag{4.4}$$

We assume the existence of an upper bound $\Xi_{\rm sw}(B, t, \log(1/\delta)) \geq \sum_{t'=1}^{t} \xi_{\rm sw}(B, t', \log(1/\delta))$ where

- $\xi_{\rm sw}(B, t, \log(1/\delta))$ is non-decreasing in B for any t, δ;
- $\Xi_{\rm sw}(B, t, \log(1/\delta))$ is concave in t for any B, δ.

Here Ψ denotes the set $\{\psi : \mathcal{B} \to \mathcal{B}\}$ which consists of all maps from actions in $\mathcal{B}$ to actions in $\mathcal{B}$. Meanwhile, for any $\theta \in \Delta_{\mathcal{B}}$, the term $\psi \diamond \theta \in \Delta_{\mathcal{B}}$ denotes the distribution over actions where $\psi \diamond \theta(b) = \sum_{b':\psi(b')=b} \theta(b')$. We note that bounded swap regret is a stronger requirement compared to bounded external regret as in (4.3), since by maximizing over a subset of functions in Ψ which map all actions in $\mathcal{B}$ to one single action, we recover the external regret by (4.4).

Assumption 2 can be satisfied by modifying many existing algorithms with external regret to the swap regret setting. In particular, we prove that the Follow-the-Regularized-Leader for swap regret (FTRL_swap) algorithm (Algorithm 20) satisfies Assumption 2 with bounds

$$\xi_{\rm sw}(B, t, \log(1/\delta)) \le \mathcal{O}(B\sqrt{H\log(B/\delta)/t}), \quad \Xi_{\rm sw}(B, t, \log(1/\delta)) \le \mathcal{O}(B\sqrt{Ht\log(B/\delta)}).$$

Both bounds have one extra $\sqrt{B}$ factor compared to the counterparts in external regret. We refer readers to Appendix C.7 for more details.

We now present the guarantees for V-learning to learn a CCE as follows. Let $A = \max_{j\in[m]} A_j$.

Theorem 25. *Suppose subroutine* ADV_BANDIT_UPDATE *satisfies Assumption 2. For any $\delta \in (0, 1)$ and $K \in \mathbb{N}$, let $\iota = \log(mHSAK/\delta)$. Choose learning rate α_t according to* (4.2) *and bonus $\{\beta_{j,t}\}_{t=1}^K$ of the j-th player so that $\sum_{i=1}^t \alpha_t^i \beta_{j,i} = \Theta(H\xi_{sw}(A_j, t, \iota) + \sqrt{H^3\iota/t})$ for any $t \in [K]$. After all the players running Algorithm 5 for K episodes, let $\widehat{\pi}_j$ be the output policy by algorithm 7 for the j-th player. Then with probability at least $1 - \delta$, the joint policy $\widehat{\pi} = \widehat{\pi}_1 \odot \ldots \odot \widehat{\pi}_m$ satisfies*

$$\max_{j\in[m]} \max_{\phi_j} [V_{j,1}^{\phi_j \diamond \widehat{\pi}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \le \mathcal{O}((H^2S/K) \cdot \Xi_{sw}(A, K/S, \iota) + \sqrt{SH^5\iota/K}).$$

When instantiating ADV_BANDIT_UPDATE *by FTRL_swap (Algorithm 20), we can* *choose $\beta_{j,t} = c \cdot A_j \sqrt{H^3 \iota / t}$ for some absolute constant c, which leads to*

$$\max_{j \in [m]} \max_{\phi_j} [V_{j,1}^{\phi_j \diamond \widehat{\pi}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \leq \mathcal{O}(A\sqrt{H^5 S \iota / K}).$$

Theorem 25 claims that, to find an $\varepsilon-$approximate CE, V-learning only needs to use a number of episodes $K = \tilde{\mathcal{O}}(H^5 S A^2 / \varepsilon^2)$, where $A = \max_{j \in [m]} A_j$. It has an extra A multiplicative factor compared to the sample complexity of finding CCE, since CE is a subset of CCE thus finding CE is expected to be more difficult. Nevertheless, the sample complexity presented here is far better than value-iteration-based algorithms, which requires at least $\Omega(H^4 S^2 (\prod_{i=1}^m A_i)/\varepsilon^2)$ episodes for finding CE Liu et al. [2021]. V-learning is also the first algorithm that breaks the curse of multiagents for finding CE in Markov games.

4.4 Monotonic V-Learning

In the previous sections, we present the V-learning algorithm whose output policy (Algorithm 7) is a nested mixture of Markov policies. Storing such an output policy requires $\mathcal{O}(HSA_jK)$ space for the j-th player. In Section 4.3, we argue this approach has a significant advantage over directly storing a general correlated policy when the number of agents is large. Nevertheless, this space complexity can be undesirable when the number of agents is small.

In this section, we introduce a simple monotonic technique to V-learning, which allows each agent to output a Markov policy when finding Nash equilibria in the two-player zero-sum setting. Storing a Markov policy only takes $\mathcal{O}(HSA_j)$ space for the j-th player. A similar result for the single-agent setting can be immediately obtained by setting the second player in the Markov game to be a dummy player with only a single action to choose from.

Monotonic update Monotonic V-learning is almost the same as V-learning with only the Line 8 in Algorithm 5 changed to

$$V_h(s_h) \leftarrow \min\{H+1-h, \tilde{V}_h(s_h), V_h(s_h)\}. \tag{4.5}$$

This step guarantees $V_h(s_h)$ to monotonically decrease at each step. This is helpful because in two-player zero-sum Markov games, all Nash equilibria share a unique value which we denote as $V^\star$. By design, we can prove that the V-values maintained in V-learning are high probability upper bounds of $V^\star$ (Lemma 96). This monotonic update allows our V-value estimates to always get closer to $V^\star$ after each update, which improves the accuracy of our V-value estimates.

Markov output policy For an arbitrary fixed $(s,h) \in \mathcal{S} \times [H]$, let t_1 be the last episode when the value $V_{1,h}(s)$ is updated (i.e., strictly decreases), and let t_2 be the last episode when the value $V_{2,h}(s)$ is updated. Then the output policy for this (s,h) has the following form.

$$\tilde{\pi}_{1,h}(\cdot|s) := \sum_{i=1}^{t_2} \alpha_{t_2}^i \pi_{1,h}^{k^i}(\cdot|s), \quad \tilde{\pi}_{2,h}(\cdot|s) := \sum_{i=1}^{t_1} \alpha_{t_1}^i \pi_{2,h}^{k^i}(\cdot|s), \tag{4.6}$$

where k^i denotes the index of episode when state s is visited at step h is visited for the i^{th} time. Recall that $\pi_{j,h}^k(\cdot|s)$ is the policy maintained by the j-th player at the beginning of the k-th episode when she runs V-learning. That is, the new output policy is simply the weighted average of policies computed in the V-learning at each (s,h) pair. Clearly, the policies $\tilde{\pi}_1$ and $\tilde{\pi}_2$ defined by (4.6) are Markov policies.

We remark that although the execution of $\tilde{\pi}_1$ and $\tilde{\pi}_2$ can be fully decentralized, in (4.6) the computation of $\tilde{\pi}_{1,h}(\cdot|s)$ depends on t_2 while the computation of $\tilde{\pi}_{2,h}(\cdot|s)$ depends on t_1. That is, two players need to communicate at the end the indexes of the most recent episodes when their V-values are updated. As a result, monotonic V-learning is not fully decentralized.

Theorem 26. *Monotonic V-learning with output policy $\tilde{\pi} = \tilde{\pi}_1 \times \tilde{\pi}_2$ as specified by*

(4.6) *has the same theoretical guarantees as Theorem 23 with the same choices of hyperparamters.*

Theorem 26 asserts that V-learning can be modified to output Markov policies when finding Nash equilibria of two-player zero-sum Markov games. As a special case, the same technique and results directly apply to the single-agent setting.

4.5 Online learning in unknown Markov Games

Model-free methods have many advantages. We have already seen one of them: the sample complexity grow mild as the number of agents is increasing and thus we aviod the **curse of mutli-agency**. In this section, we will touch another advantage of model-free learning: the learning process is fully decentralized and each agent does not need to observe the action chosen by the other agents. As a result, we can use a variant of V-learning to do online learning in unknown Markov games. Due to the hardness result presented in Section 2.5.2, we will need to work with a more modest goal. That is, to minimize the following regret against the minimax value of the game, which has appeared in prior works [Brafman and Tennenholtz, 2002, Xie et al., 2020]:

$$\mathrm{Regret}(K) := \sum\nolimits_{k=1}^{K} \big(V_1^*(s_1^k) - V_1^{\mu^k,\nu^k}(s_1^k)\big). \tag{4.7}$$

4.5.1 The V-OL algorithm

In this section, we introduce the V-OL algorithm and its regret guarantees for online learning in two-player zero-sum *unknown* Markov games. We show that not only can we achieve a sublinear regret in this challenging setting, but the regret bound can be independent of the size of the opponent's action space as well.

The V-OL algorithm. V-OL is a variant of V-learning algorithms. Bai et al. [2020] first propose V-SP as a near-optimal algorithm for the self-play setting of two-player zero-sum MGs. See the discussion at the end of this section for a detailed comparison between V-OL and V-SP.

Algorithm 8 Optimistic Nash V-learning for Online Learning (V-OL)

```
Require: Learning rate $\{\alpha_t\}_{t\geq 1}$, exploration bonus $\{\beta_t\}_{t\geq 1}$, policy update parameter $\{\eta_t\}_{t\geq 1}$
Initialize: for any $h \in [H], s \in \mathcal{S}_h, a \in \mathcal{A}_h$, $V_h(s) \leftarrow H$, $L_h(s,a) \leftarrow 0$, $N_h(s) \leftarrow 0$, $\mu_h(a|s) \leftarrow 1/|\mathcal{A}_h|$.
for episode $k = 1, \ldots, K$ do
    Receive $s_1$
    for step $h = 1, \ldots, H$ do
        Take action $a_h \sim \mu_h(\cdot|s_h)$
        Observe return $r_h$ and next state $s_{h+1}$
        Increase counter $t = N_h(s_h) \leftarrow N_h(s_h) + 1$
        $V_h(s_h) \leftarrow (1-\alpha_t)V_h(s_h) + \alpha_t(r_h + V_{h+1}(s_{h+1}) + \beta_t)$
        for all actions $a \in \mathcal{A}_h$ do
            $l_h(s_h,a) \leftarrow (H - r_h - V_{h+1}(s_{h+1}))\mathbf{I}(a_h = a)/(\mu_h(a_h|s_h) + \eta_t)$
            $L_h(s_h,a) \leftarrow (1-\alpha_t)L_h(s_h,a) + \alpha_t l_h(s_h,a)$
        Update policy $\mu$ by
            $$\mu_h(\cdot|s_h) \leftarrow \frac{\exp\{-\eta_t L_h(s_h,\cdot)/\alpha_t\}}{\sum_a \exp\{-\eta_t L_h(s_h,a)/\alpha_t\}}$$
```

In V-OL (Algorithm 8), at each time step h, the player interacts with the environment, performs an incremental update to V_h, and updates its policy μ_h. Note that the estimated value function V_h is only used for the intermediate loss $l_h(s_h,\cdot)$ in this time step, but not used in decision making. To encourage exploration in less visited states, we add a bonus term β_t. The update rule is optimistic, i.e., V_h is an upper confidence bound (UCB) on the minimax value V_h^* of the MG. Then the player samples the action according to the exponentially weighted averaged loss $L_h(s_h,\cdot)$, which is a popular decision rule in adversarial environments [Auer et al., 1995].

Intuition behind V-learning. Most existing provably efficient tabular RL algorithms learn a Q-table (table consisting of Q-values). However, since state-action pairs are necessary for updating the Q-table, for online learning in MGs, algorithms based on it inevitably require observing the opponent's actions and are thus inapplicable to unknown MGs. In contrast, V-OL does not need to maintain the Q-table at all and bypasses this challenge naturally.

Moreover, learning a Q-value function in two-player Markov games usually results in a regret or sample complexity that depends on its size SAB, whether in the self-play setting, such as VI-ULCB [Bai and Jin, 2020] and Q-SP [Bai et al., 2020], OMNI-VI-offline [Xie et al., 2020], or in the online setting, such as OMNI-VI-online [Xie et al., 2020] and Q-OL [Tian et al., 2021]. In contrast, V-learning is promising in removing the dependence on B, as formalized in Theorem 27.

Favoring more recent samples. Despite the above noted advantages of V-learning, the V-SP algorithm [Bai et al., 2020] may have a regret bound that is linear in K, as indicated by (4.9) in Theorem 27. To resolve this problem, we adopt a different set of hyperparameters to learn more aggressively by giving more weight to more recent samples. Concretely, for the self-play setting, Bai et al. [2020] specify the following hyperparameters for V-SP:

$$\alpha_t = \tfrac{H+1}{H+t},\ \beta_t = c\sqrt{\tfrac{H^4 A\iota}{t}},\ \eta_t = \sqrt{\tfrac{\log A}{At}},$$

where ι is a log factor defined later. For the online setting, we set these hyperparameters as:

$$\alpha_t = \tfrac{GH+1}{GH+t},\ \beta_t = c\sqrt{\tfrac{GH^3 A\iota}{t}},\ \eta_t = \sqrt{\tfrac{GH\log A}{At}}, \tag{4.8}$$

where $G \geq 1$ is a quantity that we tune. Ostensibly, these changes may appear small, but they are essential to attaining a sublinear regret.

Compared with $\alpha_t = {}^1\!/_t$, the learning rate $\alpha_t = {}^{H+1}\!/_{H+t}$ first proposed in [Jin et al., 2018] already favors more recent samples. Here we go one step further: our algorithm learns even more aggressively by taking $\alpha_t = {}^{GH+1}\!/_{GH+t}$ with $G \geq 1$. Moreover, we choose a larger η_t to make our algorithm care more about more recently incurred loss. β_t is set accordingly to achieve optimism.

We call this variant of V-learning V-OL, for which we prove the following regret guarantees.

Theorem 27 (Regret bounds)**.** *For any $p \in (0,1)$, let $\iota = \log({}^{HSAK}\!/_p)$. If we run* V-OL *with our hyperparameter specification* (4.8) *for some large constant c and $G \geq 1$*

in an online two-player zero-sum MG, then with probability at least $1-p$, the regret in K episodes satisfies

$$\mathrm{Regret}(K) = \mathcal{O}\big(\sqrt{GH^5SAK\iota} + KH/G + H^2S\big). \tag{4.9}$$

In particular, by taking $G = H^{-1}(K/SA)^{1/3}$ if $K \geq H^3SA$ and $G = K^{1/3}$ otherwise, with probability at least $1-p$, the regret satisfies

$$\mathrm{Regret}(K) = \begin{cases} \tilde{\mathcal{O}}\big(H^2S^{\frac{1}{3}}A^{\frac{1}{3}}K^{\frac{2}{3}} + H^2S\big), & \text{if } K \geq H^3SA, \\ \tilde{\mathcal{O}}\big(\sqrt{H^5SA}K^{\frac{2}{3}} + H^2S\big), & \text{otherwise.} \end{cases}$$

Theorem 27 shows that a sublinear regret against the minimax value of the MG is achievable for online learning in unknown MGs. As expected, the regret bound does not depend on the size of the opponent's action space B. This independence of B is particularly significant for large B, as is the case where our player plays with multiple opponents. Note that although in Theorem 27 setting the parameter G requires knowledge of K beforehand, we can use a standard doubling trick to bypass this requirement.

Remark 28. *In* V-SP *the parameter G is set to be 1. Then our choice of η_t becomes $\sqrt{H\log A/At}$, $\sqrt{H}$ times the original policy update parameter. If the other player also adopts the new $\sqrt{H\log B/Bt}$ policy update parameter, then the sample complexity of* V-SP *can actually be improved upon [Bai et al., 2020] by an H factor to $\tilde{\mathcal{O}}(H^5S(A+B)/\varepsilon^2)$.*

Comparison between V-OL and V-SP.

1. To achieve near-optimal sample complexity in the self-play setting, V-SP needs to construct upper and lower confidence bounds not only for the *minimax value* of the game, but also for the *best response* values. As a result, it uses a complicated certified policy technique, and must store the whole history of states and policies in the past K episodes for resampling. By comparing with the *minimax*

value directly, we can make V-OL provably efficient without extracting a certified policy. Therefore, V-OL only needs $\mathcal{O}(HSA)$ space instead of $\mathcal{O}(HSAK)$, and the resampling procedure is no more necessary.

2. A key feature of the proof in [Bai et al., 2020] is to make full use of a symmetric structure, which naturally arises because in the self-play setting we can control both players to follow the same learning algorithm. However, this property no longer holds for the online setting, and we must take a different proof route. Algorithmically, we need to learn more aggressively to make V-OL provably efficient.

3. V-OL also works in multi-player general-sum MGs—see Section 4.5.2.

4.5.2 Multi-player general-sum games

In this section, we extend the regret guarantees of V-OL to multi-player general-sum MGs, demonstrating the generality of our algorithm. Informally, we have the following corollary.

Corollary 29. *(informal) If we run* V-OL *with our hyperparameter specificified in* (4.8) *for our player in an online multi-player general-sum MG, then with high probability, for sufficiently large K,*

$$\mathrm{Regret}(K) = \tilde{\mathcal{O}}\big(H^2 S^{\frac{1}{3}} A^{\frac{1}{3}} K^{\frac{2}{3}} + H^2 S\big),$$

where A denotes the size of our player's action space.

The above corollary highlights the significance of removing the dependence on B in the regret bound. In particular, in a multi-player game the size of the opponents' joint action space B grows exponentially in the number of opponents, whereas the regret of V-OL only depends on the size of our player's action space A. The savings arise because V-OL bypasses the need to learn Q-tables, and the multi-player setting makes no real difference in our analysis. To formally present the construction, we need to first introduce some notation.

Consider the m-player general-sum MG

$$\mathrm{MG}_m(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathbb{P}, \{r_i\}_{i=1}^m, H), \tag{4.10}$$

where $\mathcal{S}$, H follow from the same definition in two-player zero-sum MGs, and

- for each $i \in [m]$, player i has its own action space $\mathcal{A}_i = \bigcup_{h\in[H]} \mathcal{A}_{i,h}$ and return function $r_i = \{r_{i,h} : \mathcal{S}_h \times \bigotimes_{i=1}^m \mathcal{A}_{i,h} \to [0,1]\}_{i=1}^m$, and aims to maximize its own cumulative return (here $\bigotimes$ denotes the Cartesian product of sets);
- $\mathbb{P}$ is a collection of transition functions $\{\mathbb{P}_h : \mathcal{S}_h \times \bigotimes_{i=1}^m \mathcal{A}_{i,h} \to \Delta(\mathcal{S}_{h+1})\}_{h\in[H]}$.

Like in two-player MGs, let

$$S := \sup_{h\in[H]} |\mathcal{S}_h|, \quad A_i := \sup_{h\in[H]} |\mathcal{A}_i, h| \text{ for all } i \in [m].$$

Online learning in an unknown multi-player general-sum MG can be reduced to that in a two-player zero-sum MG. Concretely, suppose we are player 1, then online learning in unknown MGs (4.10) is indistinguishable from that in the two-player zero-sum MG specified by $(\mathcal{S}, \mathcal{A}_1, \mathcal{B}, \mathbb{P}, r_1, H)$ where $\mathcal{B} = \bigotimes_{i=2}^m \mathcal{A}_i$, since we only observe and care about player 1's return. For all states $s \in \mathcal{S}_1$, define the value function using r_1 as

$$V_h^{\mu,\nu}(s) := \mathbb{E}_{\mu,\nu}[\sum\nolimits_{h'=h}^H r_{1,h'}(s_{h'}, a_{h'}, b_{h'}) | s_h = s],$$

and define the minimax value of player 1 as

$$V_1^*(s) := \max_\mu \min_\nu V_1^{\mu,\nu}(s) = \min_\nu \max_\mu V_1^{\mu,\nu}(s),$$

which is no larger than the value at the Nash equilibrium of the multi-player general-sum MG. Then we define the regret against the minimax value of player 1 as

$$\mathrm{Regret}(K) := \sum\nolimits_{k=1}^K \big(V_1^*(s_1^k) - V_1^{\mu^k,\nu^k}(s_1^k)\big).$$

We argue that this notion of regret is reasonable since we have control of only player

1 and all opponents may collude to compromise our performance. Then immediately we obtain the following corollary from Theorem 27.

Corollary 30 (Regret bound in multi-player MGs)**.** *For any* $p \in (0,1)$*, let* $\iota = \log(HSAK/p)$*. If we run* V-OL *with our hyperparameter specification* (4.8) *for some large constant* c *and the above choice of* G *for player 1 in the online multi-player general-sum MG* (4.10)*, then with probability at least* $1-p$*, the regret in* K *episodes satisfies*

$$\mathrm{Regret}(K) = \begin{cases} \tilde{\mathcal{O}}\big(H^2 S^{\frac{1}{3}} A_1^{\frac{1}{3}} K^{\frac{2}{3}} + H^2 S\big), & \text{if } K \geq H^3 S A_1, \\ \tilde{\mathcal{O}}\big(\sqrt{H^5 S A_1} K^{\frac{2}{3}} + H^2 S\big), & \text{otherwise.} \end{cases}$$

In the online informed setting, the same equivalence to a two-player zero-sum MG holds, since the other players' actions we observe can be seen as a single action $(a_i)_{i=2}^m$, and whether we observe the other players' returns does not help us decide our policies to maximize our own cumulative return. In this setting, the regret bound in [Xie et al., 2020] becomes $\tilde{\mathcal{O}}(\sqrt{H^3 S^3 \prod_{i=1}^m A_i^3 T})$, which depends exponentially on m. On the other hand, since the online informed setting has stronger assumptions than online learning in unknown MGs, the $\tilde{\mathcal{O}}(H^2 S^{1/3} A_1^{1/3} K^{2/3})$ regret bound of V-OL carries over, which has no dependence on m. This sharp contrast highlights the importance of achieving a regret independent of the size of the opponent's action space.

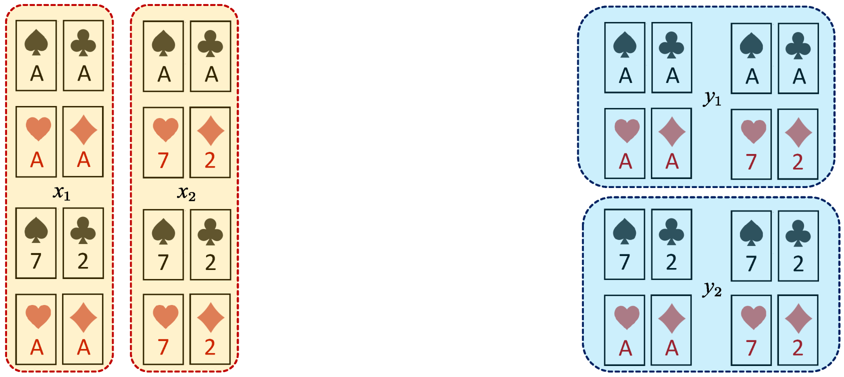
Furthermore, since in V-OL we only need to update the value function (which has HS entries), rather than update the Q-table (which has $HS \prod_{i=1}^m A_i$ entries) as in [Xie et al., 2020], we can also improve the time and space complexity by an exponential factor in m.

Chapter 5

Extensive-form Games: Prelimiaries

Imperfect Information Games—games where players can only make decisions based on partial information about the true underlying state of the game—constitute an important challenge for modern artificial intelligence. The celebrated notion of Imperfect-Information Extensive-Form games (IIEFGs) [Kuhn, 1953] offers a formulation for games with both imperfect information and sequential play. IIEFGs have been widely used for modeling real-world imperfect information games such as Poker [Heinrich et al., 2015, Moravvcík et al., 2017, Brown and Sandholm, 2018], Bridge [Tian et al., 2020], Scotland Yard [Schmid et al., 2021], etc, and achieving strong performances therein. See Figure 5-1 for an illustration of the imperfect information in the context of poker.

This section introduces the problem formulation of imperfect-information extensive-form games (IIEFGs) and the corresponding regret minimization framework. To simplify the presentation, we first introduce the standard external regret and CCE (or Nash equilibrium for 2p0s games), and subsequently discuss the more complex concepts of trigger regret and EFCE. Additionally, we provide a lower bound from Bai et al. [2022b] that characterize the minimum number of episodes required to learn the corresponding equilibria in IIEFGs.



Max-player view

Min-player view

Figure 5-1: An illustration of imperfect information through the example of poker. Here we consider a toy example with four different states. Each player is dealt either 72o or AAs. The min player's cards are on the top and the max player's cards are at the bottom.

5.1 Game Formulation

We consider two-player zero-sum IIEFGs using the formulation via Partially Observable Markov Games (POMGs), following [Kozuno et al., 2021]. In the following, $\Delta(\mathcal{A})$ denotes the probability simplex over a set $\mathcal{A}$.

Partially observable Markov games We consider finite-horizon, tabular, two-player zero-sum Markov Games with partial observability, which can be described as a tuple POMG$(H, \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$, where

- H is the horizon length;
- $\mathcal{S} = \bigcup_{h\in[H]} \mathcal{S}_h$ is the (underlying) state space with $|\mathcal{S}_h| = S_h$ and $\sum_{h=1}^{H} S_h = S$;
- $\mathcal{X} = \bigcup_{h\in[H]} \mathcal{X}_h$ is the space of information sets (henceforth *infosets*) for the *max-player* with $|\mathcal{X}_h| = X_h$ and $X := \sum_{h=1}^{H} X_h$. At any state $s_h \in \mathcal{S}_h$, the max-player only observes the infoset $x_h = x(s_h) \in \mathcal{X}_h$, where $x : \mathcal{S} \to \mathcal{X}$ is the emission function for the max-player;
- $\mathcal{Y} = \bigcup_{h\in[H]} \mathcal{Y}_h$ is the space of infosets for the *min-player* with $|\mathcal{Y}_h| = Y_h$ and $Y := \sum_{h=1}^{H} Y_h$. An infoset y_h and the emission function $y : \mathcal{S} \to \mathcal{Y}$ are defined similarly.

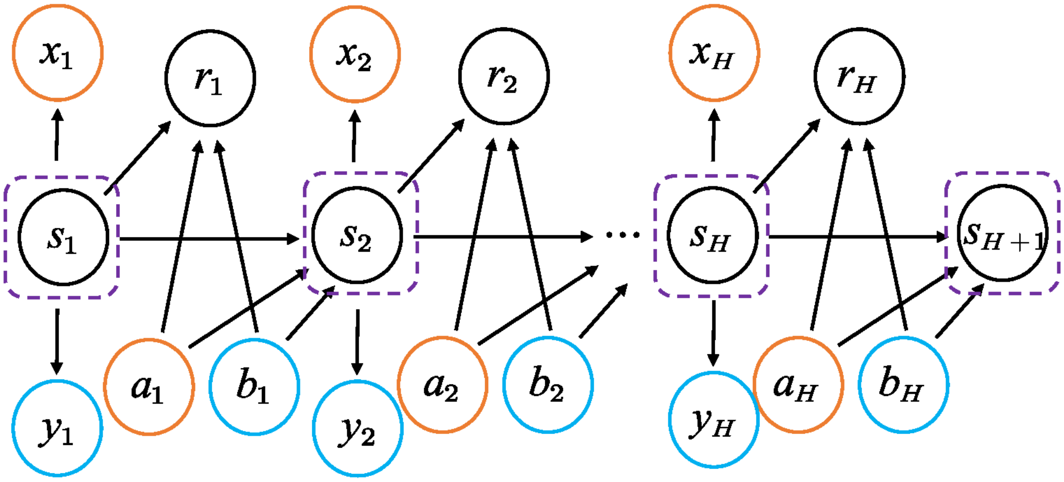


Figure 5-2: An illustration of EFGs.

- $\mathcal{A}$, $\mathcal{B}$ are the action spaces for the max-player and min-player respectively, with $|\mathcal{A}| = A$ and $|\mathcal{B}| = B$. While this assumes the action space at each infoset have equal sizes, our results can be extended directly to the case where each infoset has its own action space with (potentially) unequal sizes.

- $\mathbb{P} = \{p_0(\cdot) \in \Delta(\mathcal{S}_1)\} \cup \{p_h(\cdot|s_h, a_h, b_h) \in \Delta(\mathcal{S}_{h+1})\}_{(s_h,a_h,b_h)\in\mathcal{S}_h\times\mathcal{A}\times\mathcal{B},\ h\in[H-1]}$ are the transition probabilities, where $p_1(s_1)$ is the probability of the initial state being s_1, and $p_h(s_{h+1}|s_h, a_h, b_h)$ is the probability of transitting to s_{h+1} given state-action (s_h, a_h, b_h) at step h;

- $r = \{r_h(s_h, a_h, b_h) \in [0, 1]\}_{(s_h,a_h,b_h)\in\mathcal{S}_h\times\mathcal{A}\times\mathcal{B}}$ are the (random) reward functions with mean $\overline{r}_h(s_h, a_h, b_h)$.

See Figure 5-2 for an illustration of the decision-making process described.

Policies, value functions As we consider partially observability, each player's policy can only depend on the infoset rather than the underlying state. A policy for the max-player is denoted by $\mu = \{\mu_h(\cdot|x_h) \in \Delta(\mathcal{A})\}_{h\in[H],x_h\in\mathcal{X}_h}$, where $\mu_h(a_h|x_h)$ is the probability of taking action $a_h \in \mathcal{A}$ at infoset $x_h \in \mathcal{X}_h$. Similarly, a policy for the min-player is denoted by $\nu = \{\nu_h(\cdot|y_h) \in \Delta(\mathcal{B})\}_{h\in[H],y_h\in\mathcal{Y}_h}$. A trajectory for the max player takes the form $(x_1, a_1, r_1, x_2, \dots, x_H, a_H, r_H)$, where $a_h \sim \mu_h(\cdot|x_h)$, and the rewards and infoset transitions depend on the (unseen) opponent's actions and underlying state transition.

The overall game value for any (product) policy (μ, ν) is denoted by $V^{\mu,\nu} := \mathbb{E}_{\mu,\nu}\Big[\sum_{h=1}^{H} r_h(s_h, a_h, b_h)\Big]$. The max-player aims to maximize the value, whereas the min-player aims to minimize the value.

Tree structure and perfect recall We use a POMG with tree structure and the perfect recall assumption as our formulation for IIEFGs, following [Kozuno et al., 2021]. The class of tree-structured, perfece-recall POMGs is able to express all perfect-recall IIEFGs (defined in [Osborne and Rubinstein, 1994]) that additionally satisfy the *timeability* condition [Jakobsen et al., 2016], a mild condition that roughly requires that infosets for all players combinedly could be partitioned into ordered "layers", and is satisfied by most real-world games of interest [Kovavrík et al., 2022]. Further, our algorithms can be directly generalized to any perfect-recall IIEFG (not necessarily timeable), as we only require *each player's own game tree to be timeable* (which holds for any perfect-recall IIEFG), similar as existing OMD/CFR type algorithms [Zinkevich et al., 2007, Farina et al., 2020b]. We assume that our POMG has a *tree structure*: For any h and $s_h \in \mathcal{S}_h$, there exists a unique history $(s_1, a_1, b_1, \dots, s_{h-1}, a_{h-1}, b_{h-1})$ of past states and actions that leads to s_h. We also assume that both players have *perfect recall*: For any h and any infoset $x_h \in \mathcal{X}_h$ for the max-player, there exists a unique history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ of past infosets and max-player actions that leads to x_h (and similarly for the min-player). We further define $\mathcal{C}_{h'}(x_h, a_h) \subset \mathcal{X}_{h'}$ to be the set of all infosets in the h'-the step that are reachable from (x_h, a_h), and define $\mathcal{C}_{h'}(x_h) = \cup_{a_h \in \mathcal{A}} \mathcal{C}_{h'}(x_h, a_h)$. Finally, define $\mathcal{C}(x_h, a_h) := \mathcal{C}_{h+1}(x_h, a_h)$ as a shorthand for immediately reachable infosets.

With the tree structure and perfect recall, under any product policy (μ, ν), the probability of reaching state-action (s_h, a_h, b_h) at step h takes the form

$$\mathbb{P}^{\mu,\nu}(s_h, a_h, b_h) = p_{1:h}(s_h)\mu_{1:h}(x_h, a_h)\nu_{1:h}(y_h, b_h), \tag{5.1}$$

where we have defined the sequence-form transition probability as

$$p_{1:h}(s_h) := p_0(s_1) \prod_{h' \le h-1} p_{h'}(s_{h'+1}|s_{h'}, a_{h'}, b_{h'}),$$

where $\{s_{h'}, a_{h'}, b_{h'}\}_{h' \le h-1}$ are the histories uniquely determined from s_h by the tree structure, and the *sequence-form policies* as

$$\mu_{1:h}(x_h, a_h) := \prod_{h'=1}^{h} \mu_{h'}(a_{h'}|x_{h'}), \quad \nu_{1:h}(y_h, b_h) := \prod_{h'=1}^{h} \nu_{h'}(b_{h'}|y_{h'}),$$

where $x_{h'} = x(s_{h'})$ and $y_{h'} = y(s_{h'})$ are the infosets for the two players (with $\{x_{h'}, a_{h'}\}_{h \le h-1}$ are uniquely determined by x_h by perfect recall, and similar for $\{y_{h'}, b_{h'}\}_{h \le h-1}$).

We let $\Pi_{\max}$ denote the set of all possible policies for the max player ($\Pi_{\min}$ for the min player). In the sequence form representation, $\Pi_{\max}$ is a convex compact subset of $\mathbb{R}^{XA}$ specified by the constraints $\mu_{1:h}(x_h, a_h) \ge 0$ and $\sum_{a_h \in \mathcal{A}} \mu_{1:h}(x_h, a_h) = \mu_{1:h-1}(x_{h-1}, a_{h-1})$ for all (h, x_h, a_h), where (x_{h-1}, a_{h-1}) is the unique pair of prior infoset and action that reaches x_h (understanding $\mu_0(x_0, a_0) = \mu_0(\emptyset) = 1$).

5.2 Regret and Nash Equilibrium

We consider two standard learning goals: Regret and Nash Equilibrium. For the regret, we focus on the max-player, and assume there is an arbitrary (potentially adversarial) opponent as the min-player who may determine her policy ν^t based on all past information (including knowledge of μ^t) before the t-th episode starts. Then, the two players play the t-th episode jointly using (μ^t, ν^t). The goal for the max-player's is to design policies $\{\mu^t\}_{t=1}^{T}$ that minimizes the regret against the best fixed policy in hindsight:

$$\mathfrak{R}^T := \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \Big(V^{\mu^\dagger, \nu^t} - V^{\mu^t, \nu^t} \Big). \tag{5.2}$$

We say a product policy (μ, ν) is an ε-approximate Nash equilibrium (ε-NE) if

$$\mathrm{NEGap}(\mu, \nu) := \max_{\mu^\dagger \in \Pi_{\max}} V^{\mu^\dagger, \nu} - \min_{\nu^\dagger \in \Pi_{\min}} V^{\mu, \nu^\dagger} \leq \varepsilon,$$

i.e. μ and ν are each other's ε-approximate best response.

Definition of average policies For two-player zero-sum IIEFGs, we define the average policy of the max-player $\overline{\mu} = \frac{1}{T}\sum_{t=1}^{T} \mu^t$ (in conditional form) by

$$\overline{\mu}_h(a_h|x_h) := \frac{\sum_{t=1}^{T} \mu^t_{1:h}(x_h, a_h)}{\sum_{t=1}^{T} \mu^t_{1:h-1}(x_h)}, \tag{5.3}$$

for any h and $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$. It is straightforward to check that this $\overline{\mu}$ is exactly the averaging of μ^t in the sequence-form representation (see e.g. [Kozuno et al., 2021, Theorem 1]):

$$\overline{\mu}_{1:h}(x_h, a_h) = \frac{1}{T}\sum_{t=1}^{T} \mu^t_{1:h}(x_h, a_h) \quad \text{for all } (h, x_h, a_h). \tag{5.4}$$

Both expressions above can be used as the definition interchangably. The average policy of the min-player $\overline{\nu} = \frac{1}{T}\sum_{t=1}^{T} \nu^t$ is defined similarly.

Online-to-batch conversion It is a standard result that sublinear regret for both players ensures that the pair of average policies $(\overline{\mu}, \overline{\nu})$ is an approximate NE (see e.g. [Kozuno et al., 2021, Theorem 1]):

Proposition 31 (Regret to Nash conversion). *For any sequence of policies $\{\mu^t\}_{t=1}^{T} \in \Pi_{\max}$ and $\{\nu^t\}_{t=1}^{T} \in \Pi_{\min}$, the average policies $\overline{\mu} := \frac{1}{T}\sum_{t=1}^{T} \mu^t$ and $\overline{\nu} := \frac{1}{T}\sum_{t=1}^{T} \nu^t$ (averaged in the sequence form, cf.* (5.4)*) satisfy*

$$\mathrm{NEGap}(\overline{\mu}, \overline{\nu}) = \frac{\mathfrak{R}^T_{\max} + \mathfrak{R}^T_{\min}}{T},$$

where

$$\mathfrak{R}^T_{\max} := \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} (V^{\mu^\dagger, \nu^t} - V^{\mu^t, \nu^t}), \quad \mathfrak{R}^T_{\min} := \max_{\nu^\dagger \in \Pi_{\min}} \sum_{t=1}^{T} (V^{\mu^t, \nu^t} - V^{\mu^t, \nu^\dagger})$$

denote the regret for the two players respectively.

Therefore, an approximate NE can be learned by letting both players play some sublinear regret algorithm against each other in a self-play fashion.

Bandit feedback Throughout this paper, we consider the interactive learning (exploration) setting with bandit feedback, where the max-player determines the policy μ^t, the opponent determines ν^t (either adversarially or by running some learning algorithm, depending on the context) unknown to the max-player, and the two players play an episode of the game using policy (μ^t, ν^t). The max player observes the trajectory $(x_1^t, a_1^t, r_1^t, \dots, x_H^t, a_H^t, r_H^t)$ of her own infosets and rewards, but not the opponent's infosets, actions, or the underlying states.

Conversion to online linear regret minimization The reaching probability decomposition (5.1) implies that the value function $V^{\mu,\nu}$ is *bilinear* in (the sequence form of) (μ, ν). Thus, fixing a sequence of opponent's policies $\{\nu^t\}_{t=1}^T$, we have the linear representation

$$V^{\mu,\nu^t} = \sum_{h=1}^{H} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h}(x_h, a_h) \sum_{s_h\in x_h, b_h\in\mathcal{B}} p_{1:h}(s_h)\nu^t_{1:h}(y(s_h), b_h)\overline{r}_h(s_h, a_h, b_h).$$

Therefore, defining the *loss function* for round t as

$$\ell_h^t(x_h, a_h) := \sum_{s_h\in x_h, b_h\in\mathcal{B}} p_{1:h}(s_h)\nu^t_{1:h}(y(s_h), b_h)(1 - \overline{r}_h(s_h, a_h, b_h)) \qquad (5.5)$$

the regret $\mathfrak{R}^T$ (5.2) can be written as

$$\mathfrak{R}^T = \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^{T} \left\langle \mu^t - \mu^\dagger, \ell^t \right\rangle, \qquad (5.6)$$

where the inner product $\langle \cdot, \cdot \rangle$ is over the sequence form:

$$\left\langle \mu, \ell^t \right\rangle := \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) \ell_h^t(x_h, a_h)$$

for any $\mu \in \Pi_{\max}$.

5.3 Φ-regret minimization and the Φ-hedge algorithm

Now we introduce the Φ-regret in full generality. Consider a generic linear regret minimization problem on a *policy set* $\Pi \subset \mathbb{R}^d_{\geq 0}$ with respect to a *policy modification set* $\Phi \subset \mathbb{R}^{d\times d}$. Here Π is a convex compact subset of $\mathbb{R}^d$, and Φ is a convex compact subset of $\mathbb{R}^{d\times d}$, where each $\phi \in \Phi$ is a *policy modification function* which is a linear transformation from $\mathbb{R}^d$ to $\mathbb{R}^d$ that maps Π to itself ($\phi(\Pi) \subseteq \Pi$). For any algorithm that plays policies $\{\mu^t\}_{t=1}^T$ within T rounds and receives loss functions $\{\ell^t\}_{t=1}^T \subset \mathbb{R}^d_{\geq 0}$, the Φ-regret is defined as

$$\mathrm{Reg}^{\Phi}(T) := \sup\nolimits_{\phi\in\Phi} \left\langle \mu^t - \phi\mu^t, \ell^t \right\rangle . \tag{5.7}$$

The Φ-regret subsumes the vanilla regret (i.e. external regret) as a special case by taking Φ to be the set of all constant modifications $\Phi^{\mathsf{ext}} := \{\phi_{\mu_\star} : \mu_\star \in \Pi\}$ where $\phi_{\mu^\star}\mu = \mu^\star$ for all $\mu \in \Pi$. Another widely studied example is the *swap regret* [Blum and Mansour, 2007] (and the closely related *internal regret* [Foster and Vohra, 1998]) for normal-form games, where $\Pi = \Delta_d$ is the probability simplex over d actions, and Φ is the set of all stochastic matrices (i.e. those mapping Δ_d to itself). A primary motivation for minimizing the Φ-regret is for computing various types of *Correlated Equilibria* (CEs) in multi-player games using the online-to-batch conversion (see e.g. [Cesa-Bianchi and Lugosi, 2006]), which has been established in many games and has been a cornerstone in the online learning and games literature.

Φ-Hedge algorithm A widely used strategy for minimizing the Φ-regret is to use any (black-box) linear regret minimization algorithm on the Φ set to produce a se-

Algorithm 9 Φ-Hedge

```
Require: Finite vertex set $\Phi_0 \subset \mathbb{R}^{d\times d}$ such that $\mathrm{conv}(\Phi_0) = \Phi$; Learning rate $\eta$.
Initialize $p^1 \in \Delta_{\Phi_0}$ with $p^1_\phi = 1/|\Phi_0|$ for $\phi \in \Phi_0$.
for iteration $t = 1, \dots, T$ do
    Compute $\phi^t = \sum_{\phi\in\Phi_0} p^t_\phi \phi$.
    Set policy $\mu^t$ to be the fixed point of equation $\mu^t = \phi^t\mu^t$.
    Receive loss function $\ell^t \in \mathbb{R}^d_{\geq 0}$, suffer loss $\langle \mu^t, \ell^t\rangle$.
    Update $p^{t+1}_\phi \propto_\phi p^t_\phi \cdot \exp\{-\eta \langle \phi\mu^t, \ell^t\rangle\}$.
```

quence of $\{\phi^t\}_{t=1}^T \subset \Phi$, combined with the *fixed point technique* (e.g. [Stoltz and Lugosi, 2005])—Output policy μ^t that satisfies the fixed-point equation $\phi^t\mu^t = \mu^t$ in each round t. In the common scenario where Φ is the convex hull of a finite number of *vertices*, i.e. $\Phi = \mathrm{conv}(\Phi_0)$ where Φ_0 is a finite subset of Φ, a standard regret minimization algorithm over Φ is Hedge (a.k.a. Exponential Weights) [Arora et al., 2012], leading to the Φ-*Hedge* algorithm (Algorithm 9).

It is a standard result ([Stoltz and Lugosi, 2007], see also Lemma 118) that Algorithm 9 achieves Φ-regret bound

$$\mathrm{Reg}^\Phi(T) \leq \tfrac{\log|\Phi_0|}{\eta} + \tfrac{\eta}{2}\textstyle\sum_{t=1}^T \sum_{\phi\in\Phi_0} p^t_\phi(\langle \phi\mu^t, \ell^t\rangle)^2. \tag{5.8}$$

By choosing $\eta > 0$, this result implies a quite desirable bound

$$\mathrm{Reg}^\Phi(T) \leq L\sqrt{2\log|\Phi_0| \cdot T}$$

in the full-feedback setting (assuming bounded loss $\langle \phi\mu^t, \ell^t\rangle \leq L$), and can also be used to prove regret bounds in the bandit-feedback setting.

5.4 Extensive-form trigger regret and EFCE

To consider Φ-regret minimization in IIEFGs, it is favorable to reformuate the framework. For the purpose of this work, we consider an alternative formulation of EFGs—Tree-Form Adversarial Markov Decision Processes (TFAMDP). This model is equivalent to studying EFGs from the perspective of a single player, while treating all other players as adversaries who can change both transitions and rewards in each round. See Appendix D.3 for the details on the equivalence between TFAMDPs and IIEFGs.

5.4.1 Tree-Form Adversarial Markov Decision Processes

Tree-form adversarial MDP We consider an episodic, tabular TFAMDP which consists of the followings $(H, \mathcal{X}, \mathcal{A}, \mathcal{T}, \{p^t\}_{t\geq 1}, \{R^t\}_{t\geq 1})$. Here $H \in \mathbb{N}_+$ is the horizon length; $\mathcal{X} = \{\mathcal{X}_h\}_{h\in[H]}$ and $\mathcal{X}_h$ is the space of information sets (henceforth *infosets*) at step h with size $|\mathcal{X}_h| = X_h$ and $\sum_{h=1}^{H} X_h = X$; $\mathcal{A}$ is the action space with size $|\mathcal{A}| = A$. Next, $\mathcal{T} = \{\mathcal{C}(x,a)\}_{(x,a)\in\mathcal{X}\times\mathcal{A}}$ defines the tree structure over the infosets and actions, where $\mathcal{C}(x_h, a_h) \subset \mathcal{X}_{h+1}$ denotes the set of immediate children of (x_h, a_h). Furthermore, $\{\mathcal{C}(x_h, a_h)\}_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}$ forms a partition of $\mathcal{X}_{h+1}$. It directly follows from the tree structure of TFAMDP that the player has *perfect recall*, i.e., for any infoset $x_h \in \mathcal{X}_h$, there is a unique history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ that leads to x_h. Furthermore, $p^t = \{p_h^t\}_{h\in\{0\}\cup[H]}$; $p_0^t(\cdot) \in \Delta_{\mathcal{X}_1}$ is the initial distribution over $\mathcal{X}_1$ at episode t; $p_h^t(\cdot|x_h, a_h)$ is the transition probability from (x_h, a_h) to its immediate children $\mathcal{C}(x_h, a_h)$ at episode t; $R^t = \{R_h^t\}_{h\in[H]}$. Finally, reward $R_h^t(\cdot|x_h, a_h)$ is the distribution of the stochastic reward $r \in [0, 1]$ received at (x_h, a_h) at episode t, with expectation $\overline{R}_h^t(x_h, a_h)$.

At the beginning of episode t, an adversary will first choose the initial distribution p_0^t, transition $\{p_h^t\}_{h\in[H]}$, and reward distribution $\{R_h^t\}_{h\in[H]}$. Then in the *bandit feedback* setting, at each step h, the player observes the current infoset x_h, takes an action a_h, receives a bandit feedback of the reward $r_h^t \sim R_h^t(\cdot|x_h, a_h)$, and the environment transitions to the next state $x_{h+1} \sim p_h^t(\cdot|x_h, a_h)$.

Policies We use $\mu = \{\mu_h(\cdot|x_h)\}_{h\in[H],x_h\in\mathcal{X}_h}$ to denote a policy, where each $\mu_h(\cdot|x_h) \in \Delta_{\mathcal{A}}$ is the action distribution at infoset x_h. We say μ is a *deterministic* policy if $\mu_h(\cdot|x_h)$ takes some single action with probability 1 for any (h, x_h). Let Π denote the set of all possible policies. We denote the *sequence form* representation of policy $\mu \in \Pi$ by

$$\mu_{1:h}(x_h, a_h) := \prod_{h'=1}^{h} \mu_{h'}(a_{h'}|x_{h'}), \tag{5.9}$$

where $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ is the unique history of x_h. We also identify μ as a vector in $\mathbb{R}^{XA}_{\geq 0}$, whose (x_h, a_h)-th entry is equal to its sequence form $\mu_{1:h}(x_h, a_h)$. Let $\|\Pi\|_1 := \max_{\mu \in \Pi} \|\mu\|_1$, which admits bound $\|\Pi\|_1 \leq X$ but can in addition be smaller (See Lemma 113 for detail).

Expected loss function Given any policy μ^t at round t, the total expected loss received at round t (which equals to H minus the total rewards within round t) is given by $\langle \mu^t, \ell^t \rangle := \sum_{h, x_h, a_h} \mu^t_{1:h}(x_h, a_h) \ell^t_h(x_h, a_h)$,

where the loss function for the t-th round is given by $\ell^t = \{\ell^t_h(x_h, a_h)\}_{h, x_h, a_h} \in \mathbb{R}^{XA}_{\geq 0}$:

$$\ell^t_h(x_h, a_h) := p^t_0(x_1) \prod_{h'=1}^{h-1} p^t_{h'}(x_{h'+1} | x_{h'}, a_{h'})[1 - \overline{R}^t_h(x_h, a_h)], \tag{5.10}$$

where $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ is the unique history that leads to x_h. In the *full feedback* setting, the learner is further capable of observing the full loss vector $\ell^t \in \mathbb{R}^{XA}_{\geq 0}$ at the end of each round t.

Subtree and subtree policies For any $g \leq h$, $x_g \in \mathcal{X}_g, x_h \in \mathcal{X}_h$, and any action $a_g, a_h \in \mathcal{A}$, we say x_h or (x_h, a_h) is in the subtree rooted at x_g, written as $x_h \succeq x_g$ or $(x_h, a_h) \succeq x_g$, if x_g is either equal to x_h or is a part of the unique preceding history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ which leads to x_h. Similarly, we say x_h or (x_h, a_h) is in the subtree of (x_g, a_g), written as $x_h \succ (x_g, a_g)$ or $(x_h, a_h) \succeq (x_g, a_g)$, if (x_g, a_g) is either equal to (x_h, a_h) (only in the latter case), or is a part of the unique preceding history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ which leads to x_h.

For any $g \in [H]$, and any infoset $x_g \in \mathcal{X}_g$, we use $\mu^{x_g} = \{\mu^{x_g}_h(\cdot | x_h) \in \Delta_{\mathcal{A}} : x_h \succeq x_g\}$ to denote a subtree policy rooted at x_g.

We use Π^{x_g} and $\mathcal{V}^{x_g}$ to denote the set of all subtree policies and the set of all *deterministic* subtree policies rooted at x_g. We denote the sequence form representation of $\mu^{x_g} \in \Pi^{x_g}$ by:

$$\mu^{x_g}_{g:h}(x_h, a_h) = \begin{cases} \prod_{h'=g}^{h} \mu^{x_g}_{h'}(a_{h'} | x_{h'}) & \text{if } (x_h, a_h) \succeq x_g, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can also identify any subtree policy $\mu^{x_g} \in \Pi^{x_g}$ as a vector in $\mathbb{R}^{XA}_{\geq 0}$, whose (x_h, a_h)-th entry is equal to its sequence form $\mu^{x_g}_{g:h}(x_h, a_h)$ (which is non-zero only on the subtree rooted at x_g).

5.4.2 Extensive-form trigger regret

The notion of trigger regret is introduced in [Gordon et al., 2008, Celli et al., 2020, Farina et al., 2022a]. An *(extensive-form) trigger modification* $\phi_{x_g a_g \to m^{x_g}}$ is a policy modification that modifies any policy $\mu \in \Pi$ as follows: When x_g is visited and a_g is about to be taken (by μ), we say $x_g a_g$ is *triggered*, in which case the subtree policy rooted at x_g is then replaced by $m^{x_g} \in \Pi^{x_g}$. One can verify that the trigger modification $\phi_{x_g a_g \to m^{x_g}}$ can be written as a linear transformation that maps from Π to Π:

$$\phi_{x_g a_g \to m^{x_g}} := (I - E_{\succeq x_g a_g}) + m^{x_g} e^{\top}_{x_g a_g} \in \mathbb{R}^{XA \times XA}.$$

Here, $E_{\succeq x_g a_g}$ is a diagonal matrix with diagonal entry 1 at all (x_h, a_h) satisfying $(x_h, a_h) \succeq (x_g, a_g)$, and zero otherwise, and $e_{x_g a_g} \in \mathbb{R}^{XA}$ is an indicator vector whose only non-zero entry is 1 at (x_g, a_g). We say $\phi_{x_g a_g \to v^{x_g}}$ is a deterministic trigger modification if $v^{x_g} \in \mathcal{V}^{x_g}$ is a deterministic subtree policy. We denote the set of all deterministic trigger modifications and its convex hull as Φ^{Tr}_0 and Φ^{Tr} respectively, where

$$\Phi^{\mathsf{Tr}}_0 := \bigcup_{g, x_g, a_g} \bigcup_{v^{x_g} \in \mathcal{V}^{x_g}} \left\{ \phi_{x_g a_g \to v^{x_g}} \right\}, \qquad \Phi^{\mathsf{Tr}} = \mathrm{conv}\left\{ \Phi^{\mathsf{Tr}}_0 \right\}. \tag{5.11}$$

The *(extensive-form) trigger regret* is then defined as the difference in the total loss when comparing against the best extensive-form trigger modification in hindsight. We note that the trigger regret is a special case of Φ-regret (5.7) with $\Phi = \Phi^{\mathsf{Tr}}$.

Definition 32 (Extensive-Form Trigger Regret)**.** For any algorithm that plays policies $\mu^t \in \Pi$ at round $t \in [T]$, the extensive-form trigger regret (also the EFCE-regret) is defined as

$$\mathrm{Reg}^{\mathsf{Tr}}(T) := \max_{\phi \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \left\langle \mu^t - \phi\mu^t, \ell^t \right\rangle . \tag{5.12}$$

5.4.3 From trigger regret to Extensive-Form Correlated Equilibrium (EFCE)

The importance of extensive-form trigger regret is in its connection to computing EFCE: By standard online-to-batch conversion [Celli et al., 2020, Farina et al., 2022a], if all players have low trigger regret (with $\mathrm{Reg}_i^{\mathsf{Tr}}(T)$ for the i^{th} player), then the average joint policy $\overline{\pi}$ is an ε-EFCE, where $\varepsilon = \max_{i \in [m]} \mathrm{Reg}_i^{\mathsf{Tr}}(T)/T$. We remark in passing by taking $\Phi = \Phi^{\mathsf{ext}}$, low Φ-regret implies learning (Normal-Form) Coarse Correlated Equilibria in EFGs, as well as Nash Equilibria in the two-player zero-sum setting [Bai et al., 2022b].

Concretely, for any product policy $\pi = \{\pi_i\}_{i \in [m]}$, let $\ell^{\pi_{-i}}$ denote the expected loss function for the i^{th} player if that the other players play policy π_{-i}. We define a correlated policy $\overline{\pi}$ as a probability distribution over product policies, i.e. $\pi \sim \overline{\pi}$ gives a product policy π.

An EFCE of the game is defined as follows [Celli et al., 2020, Farina et al., 2022a].

Definition 33 (Extensive-form correlated equilibrium)**.** A correlated policy $\overline{\pi}$ is an *ε-approximate Extensive-Form Correlated Equilibrium* (EFCE) of the EFG if

$$\max_{i \in [m]} \max_{\phi \in \Phi_i^{\mathsf{EFCE}}} \mathbb{E}_{\pi \sim \overline{\pi}} \left(\langle \phi\pi_i, \ell^{\pi_{-i}} \rangle - \langle \pi_i, \ell^{\pi_{-i}} \rangle \right) \le \varepsilon.$$

We say $\overline{\pi}$ is an (exact) EFCE if the above is equality.

When the game is played with product policies for T rounds, suppose the product policy at round t is π^t, the extensive-form trigger regret (5.12) for the i^{th} player becomes

$$\mathrm{Reg}_i^{\mathsf{Tr}}(T) = \max_{\phi \in \Phi_i^{\mathsf{EFCE}}} \sum_{t=1}^{T} \left\langle \phi\pi_i^t - \pi_i^t, \ell^{\pi_{-i}^t} \right\rangle .$$

The following online-to-batch lemma for EFCE is standard, see e.g. [Celli et al., 2020].

Lemma 34 (Online-to-batch for EFCE). *Let $\{\pi^t = (\pi_i^t)_{i\in[n]}\}_{t\in[T]}$ be a sequence of product policies for all players over T rounds. Then, for the average (correlated) policy $\overline{\pi} = \mathrm{Unif}(\{\pi^t\}_{t=1}^T)$ is an ε-EFCE, where $\varepsilon = \max_{i\in[m]} \mathrm{Reg}_i^{\mathsf{Tr}}(T)/T$.*

5.5 Lower bounds

We accompany our results with information-theoretic lower bounds showing that our $\widetilde{\mathcal{O}}(\sqrt{H^3XAT})$ regret and $\widetilde{\mathcal{O}}(H^3(XA+YB)/\varepsilon^2)$ sample complexity are both near-optimal modulo $\mathrm{poly}(H)$ and log factors.

Theorem 35 (Lower bound for learning IIEFGs). *For any $A \ge 2$, $H \ge 1$, we have ($c > 0$ is an absolute constant)*

1. *(Regret lower bound) For any algorithm that controls the max player and plays policies $\{\mu^t\}_{t=1}^T$ where $T \ge XA$, there exists a game with $B = 1$ on which*

$$\mathbb{E}\big[\mathfrak{R}^T\big] = \mathbb{E}\left[\max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \left\langle \mu^t - \mu^\dagger, \ell^t \right\rangle\right] \ge c\cdot\sqrt{XAT}.$$

2. *(PAC lower bound for learning NE) For any algorithm that controls both players and outputs a final policy $(\widehat{\mu}, \widehat{\nu})$ with T episodes of play, and any $\varepsilon \in (0, 1]$, there exists a game on which the algorithm suffers from $\mathbb{E}[\mathrm{NEGap}(\widehat{\mu}, \widehat{\nu})] \ge \varepsilon$, unless*

$$T \ge c\cdot(XA+YB)/\varepsilon^2.$$

The proof of Theorem 35 (deferred to Appendix D.4) constructs a hard instance with $X = \Theta(X_H) = \Theta(A^{H-1})$ that is equivalent to A^H-armed bandit problems, and follows by a reduction to standard bandit lower bounds. We remark that our lower bounds are tight in X but did not explicitly optimize the H dependence (which is typically lower-order compared to X).

Chapter 6

Extensive-form Games: 2p0s case

A central question in IIEFGs is the problem of finding a Nash equilibrium (NE) [Nash, 1950] in a two-player zero-sum IIEFG with perfect recall. There is an extensive line of work for solving this problem with full knowledge of the game (or full feedback), by either reformulating as a linear program [Koller and Megiddo, 1992, Von Stengel, 1996, Koller et al., 1996], first-order optimization methods [Hoda et al., 2010, Kroer et al., 2015, 2018, Munos et al., 2020, Lee et al., 2021], or Counterfactual Regret Minimization [Zinkevich et al., 2007, Lanctot et al., 2009, Johanson et al., 2012, Tammelin, 2014, Schmid et al., 2019, Burch et al., 2019].

However, in the more challenging bandit feedback setting where the game is not known and can only be learned from random observations by repeated playing, the optimal sample complexity (i.e., the number of episodes required to play) for learning an NE in IIEFGs remains open. Various approaches have been proposed recently for solving this, including model-based exploration [Zhou et al., 2019, Zhang and Sandholm, 2021], Online Mirror Descent with loss estimation [Farina et al., 2021b, Kozuno et al., 2021], and Monte-Carlo Counterfactual Regret Minimization (MCCFR) [Lanctot et al., 2009, Farina et al., 2020b, Farina and Sandholm, 2021]. In a two-player zero-sum IIEFG with X, Y information sets (infosets) and A, B actions for the two players respectively, the current best sample complexity for learning an ε-approximate NE is $\widetilde{\mathcal{O}}((X^2A + Y^2B)/\varepsilon^2)$ achieved by a sample-based variant of Online Mirror Descent with implicit exploration [Kozuno et al., 2021]. However, this sample complexity scales quadratically in X, Y and still has a gap from the information-theoretic lower bound $\Omega((XA+YB)/\varepsilon^2)$ which only scales linearly. This gap is especially concerning from a practical point of view as the number of infosets is often the dominating measure of the game size in large real-world IIEFGs [Johanson, 2013].

In this Chapter, we resolve this open question by presenting the first line of algorithms that achieve $\widetilde{\mathcal{O}}((XA+YB)/\varepsilon^2)$ sample complexity for learning ε-NE in an IIEFG. The key algorithmic ingredient is thr **balanced exploration policy**, which paces learning rate in different information sets based on the scale of the subtree. Combining it with two existing framework: Online mirror descent (OMD) and counterfectual regret minimization (CFR), we develop two different algorithms achieving near-optimal sample complexity. The technique is later extended to learning CCEs in general games.

6.1 Balanced exploration policy

Our algorithms make crucial use of the following balanced exploration policies. See Figure 6-1 for an illustration of the balanced exploration policy.

Definition 36 (Balanced exploration policy)**.** For any $1 \le h \le H$, the (max-player's) *balanced exploration policy for layer h*, denoted as $\mu^{\star,h} \in \Pi_{\max}$, is defined as

$$\mu^{\star,h}_{h'}(a_{h'}|x_{h'}) := \begin{cases} \dfrac{|\mathcal{C}_h(x_{h'},a_{h'})|}{|\mathcal{C}_h(x_{h'})|}, & h' \in \{1,\dots,h-1\}, \\ 1/A, & h' \in \{h,\dots,H\}. \end{cases} \tag{6.1}$$

In words, at time steps $h' \le h-1$, the policy $\mu^{\star,h}$ plays actions proportionally to their number of descendants *within the h-th layer* of the game tree. Then at time steps $h' \ge h$, it plays the uniform policy.

Note that there are H such balanced policies, one for each layer $h \in [H]$. The balanced policy for layer $h = H$ is equivalent to the balanced strategy of Farina et al. [2020b] (cf. their Section 4.2 and Appendix A.3) which plays actions proportionally to their number descendants within the last (terminal) layer. The balanced policies for

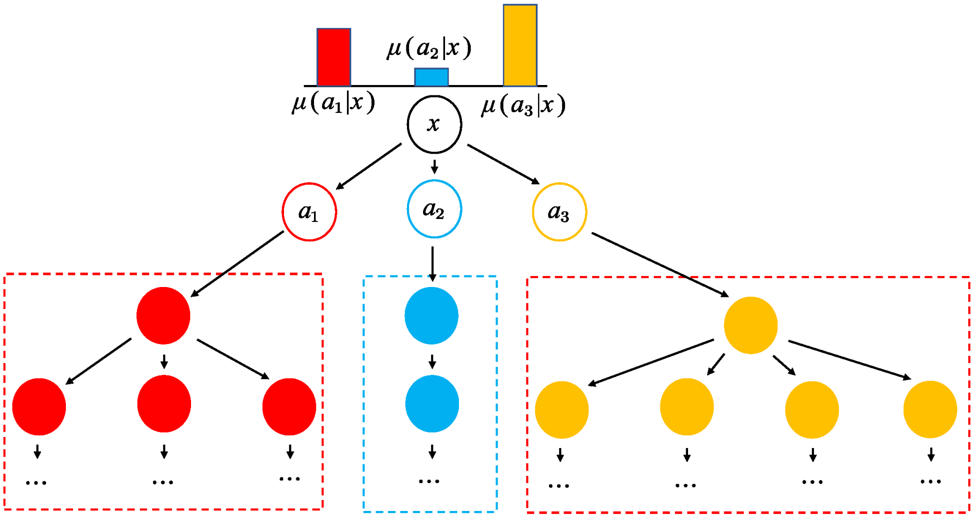


Figure 6-1: An illustration of the balanced exploration policy: sampling probability is propotional to the size of the sub game-tree rooted at the action.

layers $h \le H-1$ generalize theirs by also counting the number of descendants within earlier layers. We remark in passing that the key feature of $\mu^{\star,h}$ for our analyses is its *balancing property*, which we state in Lemma 37.

Lemma 37 (Balancing property of $\mu^{\star,h}$). *For any max-player's policy $\mu \in \Pi_{\max}$ and any $h \in [H]$, we have*

$$\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \frac{\mu_{1:h}(x_h,a_h)}{\mu^{\star,h}_{1:h}(x_h,a_h)} = X_h A.$$

Lemma 37 states that $\mu^{\star,h}$ is a good exploration policy in the sense that the distribution mismatch between it and *any* $\mu \in \Pi_{\max}$ has bounded L_1 norm. Further, the bound $X_h A$ is non-trivial—For example, if we replace $\mu^{\star,h}_{1:h}$ with the uniform policy $\mu^{\mathrm{unif}}_{1:h}(x_h,a_h) = 1/A^h$, the left-hand side can be as large as $X_h A^h$ in the worst case.

Interpretation as a transition probability We now provide an intepretation of the balanced exploration policy $\mu^{\star,h}_{1:h}$: its inverse $1/\mu^{\star,h}_{1:h}$ can be viewed as the (product) of a "transition probability" over the game tree for the max player. As a consequence, this interpretation also provides an alternative proof of Lemma 37.

For any $1 \leq h \leq H$ and $1 \leq k \leq h-1$, denote

$$p_k^{\star,h}(x_{k+1}|x_k, a_k) := |\mathcal{C}_h(x_{k+1})|/|\mathcal{C}_h(x_k, a_k)|$$

using the convention that $|\mathcal{C}_h(x_h)| = 1$. By this definition, $p_k^{\star,h}(\cdot|x_k, a_k)$ is a probability distribution over $\mathcal{C}_h(x_k, a_k)$ and can be interpreted as a balanced transition probability from (x_k, a_k) to x_{k+1}. We further denote the sequence form of the balanced transition probability by

$$p_{1:h}^{\star,h}(x_h) = \frac{|\mathcal{C}_h(x_1)|}{X_h} \prod_{k=1}^{h-1} p_k^{\star,h}(x_{k+1}|x_k, a_k) = \frac{|\mathcal{C}_h(x_1)|}{X_h} \prod_{k=1}^{h-1} \frac{|\mathcal{C}_h(x_{k+1})|}{|\mathcal{C}_h(x_k, a_k)|}. \tag{6.2}$$

Lemma 38. *For any $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$, the sequence form of the transition $p_{1:h}^{\star,h}(x_h)$ and the sequence form of balanced exploration strategy $\mu_{1:h}^{\star,h}(x_h, a_h)$ are related by*

$$p_{1:h}^{\star,h}(x_h) = \frac{1}{X_h A \cdot \mu_{1:h}^{\star,h}(x_h, a_h)}. \tag{6.3}$$

Furthermore, for any max player's policy $\mu \in \Pi_{\max}$ and any $h \in [H]$, we have

$$\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h}(x_h, a_h) p_{1:h}^{\star,h}(x_h) = 1. \tag{6.4}$$

Proof of Lemma 38 By the definition of the balanced transition probability as in Eq. (6.2) and the balanced exploration strategy as in Eq. (6.1), we have

$$\frac{1}{X_h A \cdot \mu_{1:h}^{\star,h}(x_h, a_h)} = \frac{1}{X_h A} \prod_{k=1}^{h-1} \frac{|\mathcal{C}_h(x_k)|}{|\mathcal{C}_h(x_k, a_k)|} \times A = \frac{|\mathcal{C}_h(x_1)|}{X_h} \prod_{k=1}^{h-1} \frac{|\mathcal{C}_h(x_{k+1})|}{|\mathcal{C}_h(x_k, a_k)|} = p_{1:h}^{\star,h}(x_h).$$

where the second equality used the property that $|\mathcal{C}_h(x_h)| = 1$. This proves Eq. (6.3). The proof of Eq. (6.4) is similar to the proof of Lemma 110 (a). □

Alternative proof of Lemma 37 Lemma 37 follows as a direct consequence of Eq. (6.3) and (6.4) in Lemma 38. □

Requirement on knowing the tree structure The construction of $\mu^{\star,h}$ requires knowing the number of descendants $|\mathcal{C}_h(x_{h'}, a_{h'})|$, which depends on the *structure of the game tree* for the max player. By this "structure" we refer to the parenting structure of the game tree only (which x_{h+1} is reachable from which (x_h, a_h)), not the transition probabilities and rewards.Therefore, our algorithms that use $\mu^{\star,h}$ requires knowing this tree structure beforehand. Although there exist algorithms that do not require knowing such tree structure beforehand [Zhang and Sandholm, 2021, Kozuno et al., 2021], this requirement is relatively mild as the structure can be extracted efficiently from just one tree traversal. We also remark our algorithms using the balanced policies do not impose any additional requirements on the game tree, such as the existence of a policy with lower bounded reaching probabilities at all infosets.

6.2 Online Mirror Descent

We now present our first algorithm Balanced Online Mirror Descent (Balanced OMD) and its theoretical guarantees.

6.2.1 Balanced dilated KL

At a high level, OMD algorithms work by designing loss estimators (typically using importance weighting) and solving a regularized optimization over the constraint set in each round that involves the loss estimator and a distance function as the regularizer. OMD has been successfully deployed for solving IIEFGs by using various *dilated distance generating functions* over the policy set $\Pi_{\max}$ [Kroer et al., 2015].

The main ingredient of our algorithm is the *balanced dilated KL*, a new distance measure between policies in IIEFGs.

Definition 39 (Balanced dilated KL)**.** The balanced dilated KL distance between two policies $\mu, \nu \in \Pi_{\max}$ is defined as

$$\mathrm{D}^{\mathrm{bal}}(\mu\|\nu) := \sum_{h=1}^{H} \sum_{x_h, a_h} \frac{\mu_{1:h}(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \log \frac{\mu_h(a_h|x_h)}{\nu_h(a_h|x_h)}. \tag{6.5}$$

The balanced dilated KL is a reweighted version of the *dilated KL* (a.k.a. the *dilated entropy distance-generating function*) that has been widely used for solving IIEFGs [Hoda et al., 2010, Kroer et al., 2015]:

$$\mathrm{D}(\mu\|\nu) = \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) \log \frac{\mu_h(a_h|x_h)}{\nu_h(a_h|x_h)}. \tag{6.6}$$

Compared with (6.6), our balanced dilated KL (6.5) introduces an additional reweighting term $1/\mu_{1:h}^{\star,h}(x_h, a_h)$ that depends on the balanced exploration policy $\mu^{\star,h}$ (6.1). This reweighting term is in general different for each (x_h, a_h), which at a high level will introduce a balancing effect into our algorithm. The main advantage of introducing the balanced KL is that we can give a tighter upper bound as characterized in Lemma 40 .

Lemma 40 (Bound on balanced dilated KL)**.** *Let* $\mu^{\mathrm{unif}} \in \Pi_{\max}$ *denote the uniform policy:* $\mu_h^{\mathrm{unif}}(a_h|x_h) = 1/A$ *for all* (h, x_h, a_h)*. Then we have*

$$\max_{\mu^\dagger \in \Pi_{\max}} \mathrm{D}^{\mathrm{bal}}(\mu^\dagger \| \mu^{\mathrm{unif}}) \le XA \log A.$$

Interpretation of balanced dilated KL We present an interpretation of the balanced dilated KL (6.5) as a KL distance between the reaching probabilities under the "balanced transition" (6.2) on the max player's game tree.

For any policy $\mu \in \Pi_{\max}$, we define its *balanced transition reaching probability* $\mathbb{P}_h^{\mu,\star}(x_h, a_h)$ as

$$\mathbb{P}_h^{\mu,\star}(x_h, a_h) = \mu_{1:h}(x_h, a_h) p_{1:h}^{\star,h}(x_h). \tag{6.7}$$

This is a probability measure on $\mathcal{X}_h \times \mathcal{A}$ ensured by Lemma 38. For any two probability distribution p and q, we denote $\mathrm{KL}(p\|q)$ to be their KL divergence.

Lemma 41. *For any tuple of max-player's policies $\mu, \nu \in \Pi_{\max}$, we have*

$$\mathrm{D}^{\mathrm{bal}}(\mu \| \nu) = \sum_{h=1}^{H} (X_h A) \mathrm{KL}(\mathbb{P}_h^{\mu_{1:h}, \star} \| \mathbb{P}_h^{\mu_{1:h-1}\nu_h, \star}). \tag{6.8}$$

6.2.2 Algorithm and theoretical guarantee

We now describe our Balanced OMD algorithm in Algorithm 10. Our algorithm is a variant of the IXOMD algorithm of Kozuno et al. [2021] by using the balanced dilated KL. At a high level, it consists of the following steps:

- Line 3 & 5 (Sampling): Play an episode using policy μ^t (against the opponent ν^t) and observe the trajectory. Then construct the loss estimator using importance weighting and IX bonus [Neu, 2015]:

  $$\widetilde{\ell}_h^t(x_h, a_h) := \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star, h}(x_h, a_h)}. \tag{6.9}$$

  Note that the IX bonus $\gamma \mu_{1:h}^{\star, h}(x_h, a_h)$ on the denominator makes (6.9) a slightly downward biased estimator of the true loss $\ell_h^t(x_h, a_h)$ defined in (5.5).

- Line 6 (Update policy): Update μ^{t+1} by OMD with loss estimator $\widetilde{\ell}^t$ and the balanced dilated KL distance function. Due to the sparsity of $\widetilde{\ell}^t$, this update admits an efficient implementation that updates the conditional form $\mu_h^t(\cdot | x_h)$ at the visited infoset $x_h = x_h^t$ only (described in Algorithm 23).

We are now ready to present the theoretical guarantees for the Balanced OMD algorithm.

Theorem 42 (Regret bound for Balanced OMD)**.** *Algorithm 10 with learning rate $\eta = \sqrt{XA \log A / (H^3 T)}$ and IX parameter $\gamma = \sqrt{XA\iota / (HT)}$ achieves the following regret bound with probability at least $1 - \delta$:*

$$\mathfrak{R}^T \le \mathcal{O}\Big(\sqrt{H^3 XAT\iota}\Big),$$

where $\iota := \log(3HXA/\delta)$ is a log factor.

Algorithm 10 Balanced OMD (max-player)

```
Require: Learning rate $\eta > 0$; IX parameter $\gamma > 0$.
Initialize $\mu_h^1(a_h|x_h) \leftarrow 1/A_h$ for all $(h, x_h, a_h)$.
for Episode $t = 1, \dots, T$ do
    Play an episode using $\mu^t$, observe a trajectory
        $$(x_1^t, a_1^t, r_1^t, \dots, x_H^t, a_H^t, r_H^t).$$
    for $h = H, \dots, 1$ do
        Construct loss estimator $\left\{\widetilde{\ell}_h^t(x_h, a_h)\right\}_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}$ by
            $$\widetilde{\ell}_h^t(x_h, a_h) \leftarrow \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma\mu_{1:h}^{\star,h}(x_h, a_h)}.$$
        Update policy
            $$\mu^{t+1} \leftarrow \underset{\mu\in\Pi_{\max}}{\arg\min} \left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\eta}\mathrm{D}^{\mathrm{bal}}(\mu\|\mu^t) \qquad (6.10)$$
        using the efficient implementation in Algorithm 23.
```

Letting both players run Algorithm 10, the following corollary for learning NE follows immediately from the regret-to-Nash conversion (Proposition 31).

Corollary 43 (Learning NE using Balanced OMD)**.** *Suppose both players run Algorithm 10 (and its min player's version) against each other for T rounds, with choices of η, γ specified in Theorem 42. Then, for any $\varepsilon > 0$, the average policy $(\overline{\mu}, \overline{\nu}) = (\frac{1}{T}\sum_{t=1}^T \mu^t, \frac{1}{T}\sum_{t=1}^T \nu^t)$ achieves $\mathrm{NEGap}(\overline{\mu}, \overline{\nu}) \le \varepsilon$ with probability at least $1 - \delta$, as long as the number of episodes*

$$T \ge \mathcal{O}\big(H^3(XA + YB)\iota/\varepsilon^2\big),$$

where $\iota := \log(3H(XA + YB)/\delta)$ is a log factor.

Theorem 42 and Corollary 43 are the first to achieve $\widetilde{\mathcal{O}}(\mathrm{poly}(H) \cdot \sqrt{XAT})$ regret and $\widetilde{\mathcal{O}}(\mathrm{poly}(H) \cdot (XA + YB)/\varepsilon^2)$ sample complexity for learning an ε-approximate NE for IIEFGs. Notably, the sample complexity scales only linearly in X, Y and improves significantly over the best known $\widetilde{\mathcal{O}}((X^2A + Y^2B)/\varepsilon^2))$ achieved by the

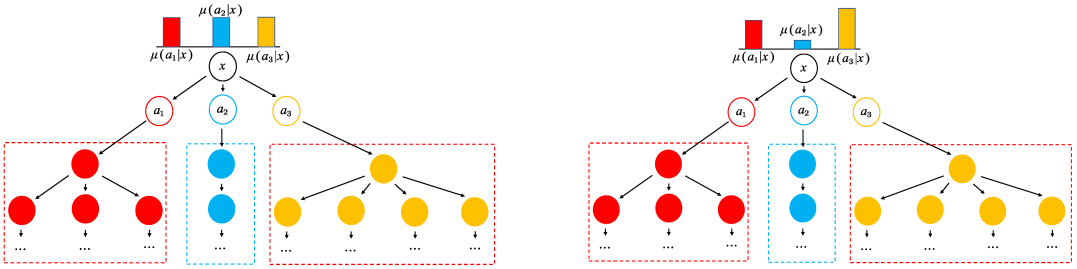


Figure 6-2: A comparison between IXOMD (left) and balanced OMD (right).

IXOMD algorithm of [Kozuno et al., 2021] by a factor of $\max\{X, Y\}$. See Figure 6-2 for an illustration of the difference between IXOMD and balanced OMD.

Overview of techniques The proof of Theorem 42 (deferred to Appendix E.3.2) follows the usual analysis of OMD algorithms where the key is to bound a distance term and an algorithm-specific "stability" like term (cf. Lemma 126 and its proof). Compared with existing OMD algorithms using the original dilated KL [Kozuno et al., 2021], our balanced dilated KL creates a "balancing effect" that preserves the distance term (Lemma 40) and shaves off an X factor in the stability term (Lemma 133 & 134), which combine to yield a $\sqrt{X}$ improvement in the final regret bound.

6.3 Counterfactual Regret Minimization

Counterfactual Regret Minimization (CFR) [Zinkevich et al., 2007] is another widely used class of algorithms for solving IIEFGs. In this section, we present a new variant Balanced CFR that also achieves sharp sample complexity guarantees. See Figure 6-3 for an illustration of CFR and its sample-based version, Mento Carlo CFR (MCCFR).

Different from OMD, CFR-type algorithms maintain a "local" regret minimizer at each infoset x_h that aims to minimize the *immediate counterfactual regret* at that infoset:

$$\mathfrak{R}_h^{\mathrm{imm},T}(x_h) := \max_{\mu \in \Delta(\mathcal{A})} \sum_{t=1}^{T} \left\langle \mu_h^t(\cdot|x_h) - \mu(\cdot), L_h^t(x_h, \cdot) \right\rangle,$$

where $L_h^t(x_h, a_h)$ is the *counterfactual loss function*

$$L_h^t(x_h, a_h) := \ell_h^t(x_h, a_h) + \sum_{h'=h+1}^{H} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h) \times \mathcal{A}} \mu_{(h+1):h'}^t(x_{h'}, a_{h'}) \ell_{h'}^t(x_{h'}, a_{h'}). \tag{6.11}$$

Controlling all the immediate counterfactual regrets $\mathfrak{R}_h^{\mathrm{imm},T}(x_h)$ will also control the overall regret of the game $\mathfrak{R}^T$, as guaranteed by the *counterfactual regret decomposition* [Zinkevich et al., 2007] (see also our Lemma 135).

6.3.1 Algorithm description

Our Balanced CFR algorithm, described in Algorithm 11, can be seen as an instantiation of the Monte-Carlo CFR (MCCFR) framework [Lanctot et al., 2009] that incorporates the balanced policies in its sampling procedure. Algorithm 11 requires regret minimization algorithms R_{x_h} for each x_h as its input, and performs the following steps in each round:

- Line 4-6 (Sampling): Play H episodes using policies $\{\mu^{t,(h)}\}_{h \in [H]}$, where each $\mu^{t,(h)} = (\mu_{1:h}^{\star,h} \mu_{h+1:H}^t)$ is a *mixture* of the balanced exploration policy $\mu^{\star,h}$ with the current maintained policy μ^t over time steps. Then, compute $\widetilde{L}_h^t(x_h, a_h)$ by (6.13) that are importance-weighted unbiased estimators of the true counterfactual loss $L_h^t(x_h, a_h)$ in (6.11).

- Line 8 (Update regret minimizers): For each (h, x_h), send the loss estimators $\{\widetilde{L}_h^t(x_h, a)\}_{a \in \mathcal{A}}$ to the local regret minimizer R_{x_h}, and obtain the updated policy $\mu_h^{t+1}(\cdot|x_h)$.

Similar as existing CFR-type algorithms, Balanced CFR has the flexibility of allowing different choices of regret minimization algorithms as R_{x_h}. We will consider two concrete instantiations of R_{x_h} as Hedge and Regret Matching in the following subsection.

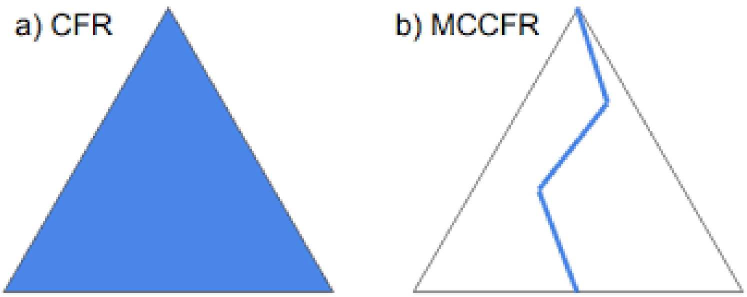


Figure 6-3: A comparison between CFR and MCCFR: instead of update the whole game tree, it only updates the policies on a trajectory.

6.3.2 Theoretical guarantee

To obtain a sharp guarantee for Balanced CFR, we first instantiate R_{x_h} as the Hedge algorithm (a.k.a. Exponential Weights, or mirror descent with the entropic regularizer; cf. Algorithm 21). Specifically, we let each R_{x_h} be the Hedge algorithm with learning rate $\eta\mu_{1:h}^{\star,h}(x_h, a)$.Note that this quantity depends on x_h but not a. With this choice, Line 8 of Algorithm 11 takes the following explicit form:

$$\mu_h^{t+1}(a|x_h) \propto_a \mu_h^t(a|x_h) \cdot e^{-\eta\mu_{1:h}^{\star,h}(x_h,a)\cdot \widetilde{L}_h^t(x_h,a)}. \tag{6.12}$$

We are now ready to present the theoretical guarantees for the Balanced CFR algorithm.

Theorem 44 ("Regret" bound for Balanced CFR)**.** *Suppose the max player plays Algorithm 11 where each R_{x_h} is instantiated as the Hedge algorithm* (6.12) *with $\eta = \sqrt{XA\iota/(H^3T)}$. Then, the policies μ^t achieve the following "regret" bound with probability at least $1-\delta$:*

$$\widetilde{\mathfrak{R}}^T := \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^{T} \left\langle \mu^t - \mu^\dagger, \ell^t \right\rangle \le \mathcal{O}(\sqrt{H^3XAT\iota}),$$

where $\iota = \log(10XA/\delta)$ is a log factor.

The $\widetilde{\mathcal{O}}(\sqrt{H^3XAT})$ "regret" achieved by Balanced CFR matches that of Balanced OMD. However, we emphasize that the quantity $\widetilde{\mathfrak{R}}^T$ is *not strictly speaking a regret,*

Algorithm 11 Balanced CFR (max-player)

```
Require: Regret minimization algorithm $R_{x_h}$ for all $(h, x_h)$.
Initialize policy $\mu_h^1(a_h|x_h) \leftarrow 1/A$ for all $(h, x_h, a_h)$.
for round $t = 1, \dots, T$ do
    for $h = 1, \dots, H$ do
        Set policy $\mu^{t,(h)} \leftarrow (\mu_{1:h}^{\star,h} \mu_{h+1:H}^t)$.
        Play an episode using $\mu^{t,(h)} \times \nu^t$, observe a trajectory
            $$(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \cdots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)}).$$
        Compute loss estimators for all $(h, x_h, a_h)$:
            $$\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\left\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\right\}}{\mu_{1:h}^{\star,h}(x_h, a_h)} \left(H - h + 1 - \sum_{h'=h}^{H} r_{h'}^{t,(h)}\right). \quad (6.13)$$
    for all $h \in [H]$ and $x_h \in \mathcal{X}_h$ do
        Update the regret minimizer at $x_h$ and obtain policy:
            $$\mu_h^{t+1}(\cdot|x_h) \leftarrow R_{x_h}.\textsc{Update}(\{\widetilde{L}_h^t(x_h, a)\}_{a \in \mathcal{A}}). \quad (6.14)$$
```

as it measures performance of the policy $\{\mu^t\}$ *maintained* in the Balanced CFR algorithm, not the sampling policy $\mu^{t,(h)}$ that the Balanced CFR algorithm have *actually played.* Nevertheless, we remark that such a form of "regret" bound is the common type of guarantee for all existing MCCFR type algorithms [Lanctot et al., 2009, Farina et al., 2020b].

Self-play of Balanced CFR Balanced CFR can also be turned into a PAC algorithm for learning ε-NE, by letting the two players play Algorithm 11 against each other for T rounds of self-play using the following protocol: Within each round, the max player plays policies $\left\{\mu^{t,(h)}\right\}_{h=1}^{H}$ while the min player plays the fixed policy ν^t; then symmetrically the min player plays $\left\{\nu^{t,(h)}\right\}_{h=1}^{H}$ while the max player plays the fixed μ^t. Overall, each round plays $2H$ episodes.

Theorem 44 directly implies the following corollary for the above self-play algorithm on learning ε-NE, by the regret-to-Nash conversion (Proposition 31).

Corollary 45 (Learning NE using Balanced CFR)**.** *Let both players play Algo-*

rithm 11 in a self-play fashion against each other for T rounds, where each R_{x_h} is instantiated as the Hedge algorithm (6.12) *with η specified in Theorem 44. Then, for any $\varepsilon > 0$, the average policy $(\overline{\mu}, \overline{\nu}) = (\frac{1}{T}\sum_{t=1}^{T}\mu^t, \frac{1}{T}\sum_{t=1}^{T}\nu^t)$ achieves $\mathrm{NEGap}(\overline{\mu}, \overline{\nu}) \le \varepsilon$ with probability at least $1-\delta$, as long as*

$$T \ge \mathcal{O}(H^3(XA + YB)\iota/\varepsilon^2),$$

where $\iota := \log(10(XA+YB)/\delta)$ is a log factor. The total amount of episodes played is at most

$$2H \cdot T = \mathcal{O}(H^4(XA + YB)\iota/\varepsilon^2).$$

Corollary 45 shows that Balanced CFR requires $\widetilde{\mathcal{O}}(H^4(XA + YB)/\varepsilon^2)$ episodes for learning an ε-NE, which is H times larger than Balanced OMD but otherwise also near-optimal with respect to the lower bound (Theorem 35) modulo an $\widetilde{\mathcal{O}}(\mathrm{poly}(H))$ factor. This improves significantly over the current best sample complexity achieved by CFR-type algorithms, which are either $\mathrm{poly}(X, Y, A, B)/\varepsilon^4$ [Farina and Sandholm, 2021], or potentially $\mathrm{poly}(X, Y, A, B)/\varepsilon^2$ using the MCCFR framework of [Lanctot et al., 2009, Farina et al., 2020b] but without any known such instantiation.

Overview of techniques The proof of Theorem 44 (deferred to Appendix E.4.2) follows the usual analysis pipeline for MCCFR algorithms that decomposes the overall regret $\widetilde{\mathfrak{R}}_T$ into combinations of immediate counterfactual regrets $\mathfrak{R}_h^{\mathrm{imm},T}(x_h)$, and bounds each by regret bounds (of the regret minimizer R_{x_h}) plus concentration terms. We adopt a sharp application of this pipeline by using a tight counterfactual regret decomposition (Lemma 135), as well as using the balancing property of $\mu^{\star,h}$ which yields sharp bounds on both the regret and concentration terms (Lemma 136-138).

We remark that our techniques can also be used for analyzing CFR type algorithms in the full-feedback setting. Concretely, we provide a sharp $\mathcal{O}(\sqrt{H^3\|\Pi_{\max}\|_1 \log A \cdot T})$ regret bound for a “vanilla" CFR algorithm in the full full-feedback setting, matching the result of [Zhou et al., 2020, Lemma 2].

6.3.3 Balanced CFR with regret matching

Many real-world applications of CFR-type algorithms use Regret Matching [Hart and Mas-Colell, 2000] instead of Hedge as the regret minimizer, due to its practical advantages such as learning-rate free and pruning effects [Tammelin, 2014, Burch et al., 2019].

In this section, we show that Balanced CFR instantiated with Regret Matching enjoys $\widetilde{\mathcal{O}}(\sqrt{H^3XA^2T})$ "regret" and $\widetilde{\mathcal{O}}((H^4(XA^2+YB^2)/\varepsilon^2)$ sample complexity for learning ε-NE (Theorem 46 & Corollary 47). The sample complexity is also sharp in X, Y, though is A (or B) times worse than the Hedge version, which is expected due to the difference between the regret minimizers.

We consider instantiating Line 8 of Algorithm 11 using the following Regret Matching algorithm

$$\begin{aligned} &\mu_h^{t+1}(a|x_h) = \frac{\left[R_{x_h}^t(a)\right]_+}{\sum_{a'\in\mathcal{A}}\left[R_{x_h}^t(a')\right]_+}, \\ &\text{where} \quad R_{x_h}^t(a) := \sum_{\tau=1}^{t}\left\langle \mu_h^\tau(\cdot|x_h), \widetilde{L}_h^\tau(x_h,\cdot)\right\rangle - \widetilde{L}_h^\tau(x_h,a) \quad \text{for all } a\in\mathcal{A}. \end{aligned} \tag{6.15}$$

as the regret minimization sub-routine for each information set. See Algorithm 12 for the full version.

We now present the main theoretical guarantees for Balanced CFR with regret matching. The proof of Theorem 46 can be found in Section E.4.6.

Theorem 46 ("Regret" bound for Balanced CFR with Regret Matching)**.** *Suppose the max player plays Algorithm 11 where each R_{x_h} is instantiated as the Regret Matching algorithm* (6.15)*. Then the policies μ^t achieve the following regret bound with probability at least $1-\delta$:*

$$\widetilde{\mathfrak{R}}^T := \max_{\mu^\dagger\in\Pi_{\max}}\sum_{t=1}^{T}\left\langle \mu^t-\mu^\dagger, \ell^t\right\rangle \le \mathcal{O}(\sqrt{H^3XA^2T\iota}),$$

where $\iota = \log(10XA/\delta)$ is a log factor. Further, each round plays H episodes against ν^t (so that the total number of episodes played is HT).

Algorithm 12 Balanced CFR with Regret Matching (max player)

```
Require: Learning rate $\eta > 0$.
Initialize policies $\mu_h^1(a_h|x_h) \leftarrow 1/A$ for all $(h, x_h, a_h)$.
for iteration $t = 1, \dots, T$ do
    for $h = 1, \dots, H$ do
        Set policy $\mu^{t,(h)} \leftarrow (\mu^{\star,h}_{1:h}\mu^t_{h+1:H})$.
        Play an episode using $\mu^{t,(h)} \times \nu^t$, observe trajectory
            $(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \cdots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)})$.
        Compute loss estimators for all $(h, x_h, a_h)$:
            $\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\left\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\right\}}{\mu^{\star,h}_{1:h}(x_h, a_h)} \times \left(H - h + 1 - \sum_{h'=h}^{H} r_{h'}^{t,(h)}\right)$.
    for all $h \in [H]$ and $x_h \in \mathcal{X}_h$ do
        Update policy at $x_h$ using Regret Matching:
            $\mu_h^{t+1}(a|x_h) = \frac{\left[R^t_{x_h}(a)\right]_+}{\sum_{a' \in \mathcal{A}} \left[R^t_{x_h}(a')\right]_+}$,
            where $R^t_{x_h}(a) := \sum_{\tau=1}^{t} \left\langle \mu_h^\tau(\cdot|x_h), \widetilde{L}_h^\tau(x_h, \cdot) \right\rangle - \widetilde{L}_h^\tau(x_h, a)$ for all $a \in \mathcal{A}$.
```

We then have the following corollary directly by the regret-to-Nash conversion (Proposition 31).

Corollary 47 (Learning Nash using Balanced CFR with Regret Matching)**.** *Letting both players play Algorithm 11 in a self-play fashion against each other for T rounds, where each R_{x_h} is instantiated as the Regret Matching algorithm* (6.15)*. Then, for any $\varepsilon > 0$, the average policy $(\overline{\mu}, \overline{\nu}) = (\frac{1}{T}\sum_{t=1}^T \mu^t, \frac{1}{T}\sum_{t=1}^T \nu^t)$ achieves* $\mathrm{NEGap}(\overline{\mu}, \overline{\nu}) \le \varepsilon$ *with probability at least $1 - \delta$, as long as*

$$T \ge \mathcal{O}(H^3(XA^2 + YB^2)\iota/\varepsilon^2),$$

where $\iota := \log(10(XA + YB)/\delta)$ *is a log factor. The total amount of episodes played* *is at most*

$$2H \cdot T = \mathcal{O}(H^4(XA^2 + YB^2)\iota/\varepsilon^2).$$

6.4 Extension to multi-player games

In this section, we show that our Balanced OMD and Balanced CFR generalize directly to learning Coarse Correlated Equilibria (CCE) in multi-player general-sum games.

We consider an m-player general-sum IIEFG with X_i infosets and A_i actions for the i-th player. Let V_i denote the game value (expected cumulative reward) for the i-th player.

Definition 48 (NFCCE)**.** A joint policy π (for all players) is an ε-approximate Normal-Form Coarse Correlated Equilibrium (NFCCE) if

$$\mathrm{CCEGap}(\pi) := \max_{i\in[m]} \Big(\max_{\pi_i^\dagger \in \Pi_i} V_i^{\pi_i^\dagger, \pi_{-i}} - V_i^{\pi} \Big) \le \varepsilon,$$

i.e., no player can gain more than ε in her own reward by deviating from π and playing some other policy on her own.

We remark that the NFCCE differs from other types of Coarse Correlated Equilibria in the literature such as the EFCCE [Farina et al., 2020a]. Such distinctions only exist for (Coarse) Correlated Equilibria and not for the NE studied in the previous sections. Using the known connection between no-regret and NFCCE [Celli et al., 2019a], we can learn an ε-NFCCE in an multi-player IIEFG sample-efficiently by letting all players run either Balanced CFR or Balanced OMD in a self-play fashion. In the following, we let $\{\pi_i^t\}_{t=1}^T$ denote the policies maintained by player i, and $\pi^t := \prod_{i=1}^m \pi_i^t$ denote their joint policy in the t-th round.

Theorem 49 (Learning NFCCE sample-efficiently using Balanced OMD / Balanced CFR)**.** *We have*

1. *(Balanced OMD) Let all players play Algorithm 10 for T rounds with learning rate $\eta = \sqrt{X_i A_i \log A_i / (H^3 T)}$ and IX parameter $\gamma = \sqrt{X_i A_i \iota / (HT)}$ for the i-th player. Then for any $\varepsilon > 0$, the average policy $\overline{\pi}$ uniformly sampled from $\{\pi^t\}_{t=1}^T$ satisfies $\mathrm{CCEGap}(\overline{\pi}) \le \varepsilon$ with probability at least $1 - \delta$, as long as the number of episodes*

   $$T \ge \mathcal{O}\Big(H^3 \iota \Big(\max_{i \in [m]} X_i A_i \Big) / \varepsilon^2 \Big),$$

   where $\iota := \log(3H \sum_{i=1}^m X_i A_i / \delta)$ is a log factor.

2. *(Balanced CFR) Let all players play Algorithm 11 in the same self-play fashion as Corollary 45 for T rounds, with R_{x_h} instantiated as Hedge (6.12) with learning rate $\eta = \sqrt{X_i A_i \iota / (H^3 T)}$ for the i-th player. Then for any $\varepsilon > 0$, the average policy $\overline{\pi}$ uniformly sampled from $\{\pi^t\}_{t=1}^T$ satisfies $\mathrm{CCEGap}(\overline{\pi}) \le \varepsilon$ with probability at least $1 - \delta$, as long as $T \ge \mathcal{O}\big(H^3 \iota (\max_{i \in [m]} X_i A_i) / \varepsilon^2\big)$. The total number of episodes played is at most*

   $$mH \cdot T = \mathcal{O}\Big(H^4 m \iota \cdot \Big(\max_{i \in [m]} X_i A_i \Big) / \varepsilon^2 \Big).$$

   where $\iota := \log(10 \sum_{i=1}^m X_i A_i / \delta)$ is a log factor.

For both algorithms, the number of episodes for learning an ε-NFCCE scales linearly with $\max_{i \in [m]} X_i A_i$ (with Balanced CFR having an additional Hm factor than Balanced OMD), compared to the best existing $\max_{i \in [m]} X_i^2 A_i$ dependence (e.g. by self-playing IXOMD [Kozuno et al., 2021]). The proof of Theorem 49 is in Appendix E.5.1.

Chapter 7

Extensive-form Games: general case

In multi-player general-sum EFGs, computing an approximate Nash equilibrium (NE) is PPAD-hard [Daskalakis et al., 2009] and thus likely intractable. A reasonable and computationally tractable solution concept in general-sum EFGs is the *extensive-form correlated equilibria* (EFCE) [Von Stengel and Forges, 2008, Gordon et al., 2008, Celli et al., 2020, Farina et al., 2022a]. It is known that, as long as each player runs an uncoupled dynamics minimizing a suitable EFCE-regret, their average joint policy will converge to an EFCE [Greenwald and Jafari, 2003].

Existing algorithms of minimizing the EFCE-regret are mostly built upon the *regret decomposition* techniques [Zinkevich et al., 2007], which utilize the structure of the game and the set of policy modifications [Celli et al., 2020, Morrill et al., 2021, Farina et al., 2022a, Song et al., 2022b]. For example, Morrill et al. [2021] decomposes the EFCE-regret to local regrets at each information set (infoset) with each of them handled by a local regret minimizer; Farina et al. [2022a] utilizes the trigger structure of the policy modification set to decompose the regret to external-like regrets.

There are at least two alternative approaches to designing regret minimization algorithms for EFGs.

- The first is to convert a EFG to a normal-form game (NFG) and use NFG-based algorithms such as Φ-Hedge [Greenwald and Jafari, 2003]. This approach typically admits simple algorithm designs and sharp regret bounds by directly

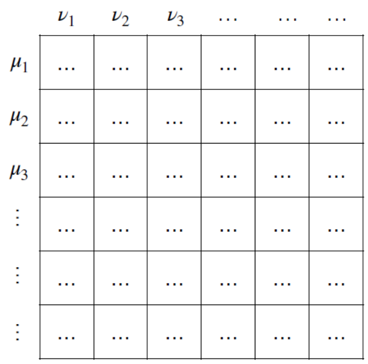


Figure 7-1: The naive reduction from EFGs to NFGs induces a exponential blow-up.

translating existing results in NFGs [Stoltz and Lugosi, 2007]. However, the conversion introduces an exponential blow-up in the game size, and makes such algorithms computationally intractable in general (See Figure 7-1 for an illustration). The computational efficiency of these NFG-based algorithms is recently investigated by Farina et al. [2022b] in the external regret minimization problem, who provided an efficient implementation of an NFG-based algorithm using "kernel tricks".

- The second is to use Online Mirror Descent (OMD) algorithms via suitably designed regularizers over the parameter space. This approach has been successfully implemented in minimizing the external regret [Kroer et al., 2015] but not yet generalized to the EFCE-regret, as it remains unclear how to design suitable regularizers for the policy modification space.

In this chapter, we develop the first line of EFCE-regret minimization algorithms along both lines of approaches above, and identify an equivalence between them. We consider EFCE-regret minimization in EFGs with X infosets, A actions, and maximum L_1-norm of sequence-form policies bounded by $\|\Pi\|_1$ (cf. Section 5.4 for the formal definition).

7.1 Efficient Φ-Hedge for Trigger Regret Minimization

In this section, we study the Φ-Hedge algorithm (Algorithm 9) for minimizing the trigger regret. Naively, Algorithm 9 requires maintaining and updating $p^t \in \Delta_{\Phi_0}$ (cf. Line 6), whose computational cost is linear in $|\Phi_0^{\mathsf{Tr}}|$ which can be exponential in X in the worst case. Notice $|\Phi_0^{\mathsf{Tr}}|$ is at least the number of deterministic policies of the game, which could be $A^{O(X)}$ in the worst case. We begin by deriving an efficient implementation of the iterate $\phi^t \in \Phi$ (of Line 3) directly by exploiting the structure of Φ_0^{Tr}.

7.1.1 Efficient implementation of Φ^{Tr}-Hedge algorithm

We first use a standard trick to convert the computation of ϕ^t (Line 3 & 6, Algorithm 9) in Φ-Hedge to evaluating the gradient of a suitable log-partition function. This is stated in the lemma below (for any generic Φ_0), whose proof can be found in Appendix F.1.2.

Lemma 50 (Conversion to log-partition function). *Define the log-partition function $F^{\Phi_0} : \mathbb{R}^{d\times d} \to \mathbb{R}$*

$$F^{\Phi_0}(M) := \log \sum_{\phi\in\Phi_0} \exp\{-\langle \phi, M\rangle\}. \tag{7.1}$$

Then Line 3 of Φ-Hedge (Algorithm 9) has a closed-form update for all $t \geq 1$:

$$\phi^t = -\nabla F^{\Phi_0}\left(\eta \sum_{s=1}^{t-1} M^s\right) = -\frac{\sum_{\phi\in\Phi_0} \exp\left\{ -\eta \left\langle \phi, \sum_{s=1}^{t-1} M^s \right\rangle \right\}\phi}{\sum_{\phi\in\Phi_0} \exp\left\{ -\eta \left\langle \phi, \sum_{s=1}^{t-1} M^s \right\rangle \right\}}, \qquad M^t := \ell^t(\mu^t)^\top. \tag{7.2}$$

Eq. (7.2) suggests a strategy for evaluating $\phi^t = -\nabla F^{\Phi_0}(\eta \sum_{s=1}^{t-1} M^s)$—So long as the vertex set Φ_0 has some structure that allows efficient evaluation of the sum of exponentials on the numerators and denominators (i.e. faster than naive sum), ϕ^t

may be computed directly in sublinear in $|\Phi_0|$ time, and there is no need to maintain the underlying distribution $p^t \in \Delta_{\Phi_0}$.

The following lemma enables such an efficient computation for the log-partition function $F^{\mathsf{Tr}} := F^{\Phi^{\mathsf{Tr}}}$ (and its gradient) associated with the trigger modification set $\Phi = \Phi^{\mathsf{Tr}}$. This lemma (proof deferred to Appendix F.1.3) is a consequence of the specific structure of Φ_0 (cf. (5.11)), whose elements are indexed by a sequence $x_g a_g$ and a deterministic subtree policy $v^{x_g} \in \mathcal{V}^{x_g}$.

Lemma 51 (Recursive expression of F^{Tr} and ∇F^{Tr})**.** *For any loss matrix* $M \in \mathbb{R}^{XA \times XA}$*, the* EFCE log-partition function *can be written as*

$$F^{\mathsf{Tr}}(M) = \log \textstyle\sum_{g, x_g, a_g} \exp\Big\{ - \big\langle I - E_{\succeq x_g a_g}, M \big\rangle + F_{x_g a_g, x_g}(M) \Big\}, \tag{7.3}$$

where for any $x_h \succeq x_g$*,*

$$F_{x_g a_g, x_h}(M) := \log \textstyle\sum_{a_h} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_g a_g, x_{h+1}}(M) \Big\}. \tag{7.4}$$

Furthermore, define $\lambda = (\lambda_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \Delta_{XA}$ *and* $m = (m_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}}$ *with* $m_{x_g a_g} \in \Pi^{x_g}$ *(and also identified as a vector in* $\mathbb{R}^{XA}$*) as*

$$\lambda_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ - \big\langle I - E_{\succeq x_g a_g}, M \big\rangle + F_{x_g a_g, x_g}(M) \Big\}, \tag{7.5}$$

$$m_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \textstyle\sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_g a_g, x_{h+1}}(M) \Big\}, \tag{7.6}$$

then we have

$$-\nabla F^{\mathsf{Tr}}(M) = \phi(\lambda, m) := \textstyle\sum_{g, x_g, a_g} \lambda_{x_g a_g} (I - E_{\succeq x_g a_g} + m_{x_g a_g} e_{x_g a_g}^{\top}). \tag{7.7}$$

Above, $\lambda = (\lambda_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \Delta_{XA}$ is a probability distribution over $\mathcal{X} \times \mathcal{A}$, and $m = (m_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \mathcal{M} \equiv \prod_{g, x_g a_g} \Pi^{x_g a_g}$ is a collection of subtree policies $m_{x_g a_g}$, where each $m_{x_g a_g} \in \Pi^{x_g}$ is a subtree policy that specifies an action distribution $m_{x_g a_g, h}(a_h | x_h)$ for every $x_h \succeq x_g$, and can be identified with a vector in $\mathbb{R}^{XA}$ (c.f. Section 5.4).

Algorithm 13 EFCE-OMD (FTRL form; equivalent OMD form in Algorithm 24)

```
Require: Learning rate $\eta > 0$.
1: for $t = 1, 2, \dots, T$ do
2:     For each $x_g a_g \in \mathcal{X} \times \mathcal{A}$, from the reverse order of $x_h$, compute $m^t_{x_g a_g, h}(a_h|x_h)$ and $F^t_{x_g a_g, x_h}$
           $m^t_{x_g a_g,h}(a_h|x_h) \propto_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \Big\},$   (7.8)
           $F^t_{x_g a_g, x_h} = \log \sum_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \Big\},$   (7.9)
3:     Compute $\lambda^t_{x_g a_g}$ as
           $\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ -\eta \big\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t-1} M^s \big\rangle + F^t_{x_g a_g, x_g} \Big\}.$   (7.10)
4:     Compute $\phi^t = \phi(\lambda^t, m^t)$ where $\phi$ is in Eq. (7.7).
5:     Compute the policy $\mu^t$, which is a solution of the fixed point equation $\phi^t \mu^t = \mu^t$.
6:     Receive loss $\ell^t = \{\ell^t_h(x_h, a_h)\}_{(x_h, a_h) \in \mathcal{X} \times \mathcal{A}} \in \mathbb{R}^{XA}_{\geq 0}$.
7:     Compute matrix loss $M^t = \ell^t (\mu^t)^\top \in \mathbb{R}^{XA \times XA}_{\geq 0}$.
```

The recursive structure in Lemma 51 offers a roadmap for evaluating (λ, m) and thus $\nabla F^{\mathsf{Tr}}(M)$ in $O(X^2 A^2)$ time (formal statement in Appendix F.1.4). Applying Lemma 51 with $M = \eta \sum_{s=1}^{t-1} M^s$ gives an efficient implementation of (7.2), i.e. the Φ-Hedge algorithm with $\Phi = \Phi^{\mathsf{Tr}}$. For clarity, we summarize this in Algorithm 13. We remark that the parameters (λ^t, m^t) therein can also be expressed in terms of (λ^{t-1}, m^{t-1}) and M^{t-1}, which we present in Algorithm 24 (the equivalent "OMD" form) in Appendix F.1.1. We also note that the fixed point equation $\phi^t \mu = \mu$ in Line 5 can be solved in $O(X^2 A^2)$ time [Farina et al., 2022a, Corollary 4.15].

7.1.2 Equivalence to FTRL and OMD

We now show that Algorithm 13 is equivalent to FTRL and OMD with suitable dilated entropies and divergences (hence the name EFCE-OMD). To achieve this goal we first need to introduce the subtree dilated entropy and subtree dilated KL divergence, a variant of dilated entropy and dilated KL divergence introduced in Hoda et al. [2010], Kroer et al. [2015], Kozuno et al. [2021]. Here the modification we made is that we allow these quantities to be rooted at any infoset x_g. The original version is recovered when the choose the full game tree as the subtree. These quantities were used to define the trigger dilated entropy and trigger dilated KL divergence as in Section 7.1.2.

Definition 52 (Dilated entropy and Dilated KL divergence)**.** The dilated entropy H_{x_g} rooted at x_g of subtree policy $\mu^{x_g} \in \Pi^{x_g}$ is defined as

$$H_{x_g}(\mu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \mu^{x_g}_{g:h}(x_h,a_h) \log \mu^{x_g}_{h}(a_h|x_h). \tag{7.11}$$

The dilated KL divergence D_{x_g} rooted at x_g between two subtree policies $\mu^{x_g}, \nu^{x_g} \in \Pi^{x_g}$ is defined as

$$D_{x_g}(\mu^{x_g}\|\nu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \mu^{x_g}_{g:h}(x_h,a_h) \log \frac{\mu^{x_g}_{h}(a_h|x_h)}{\nu^{x_g}_{h}(a_h|x_h)}. \tag{7.12}$$

We define the trigger dilated entropy function and trigger dilated KL divergence function over $(\lambda, m) \in \Delta_{XA} \times \mathcal{M}$ as

$$\begin{aligned} H^{\mathsf{Tr}}(\lambda, m) &:= H(\lambda) + \textstyle\sum_{g,x_g,a_g} \lambda_{x_g a_g} H_{x_g}(m_{x_g a_g}), \\ D^{\mathsf{Tr}}(\lambda, m\|\lambda', m') &:= D_{\mathrm{KL}}(\lambda\|\lambda') + \textstyle\sum_{g,x_g,a_g} \lambda_{x_g a_g} D_{x_g}(m_{x_g a_g}\|m'_{x_g a_g}), \end{aligned}$$

where $H(\cdot)$ and $D_{\mathrm{KL}}(\cdot\|\cdot)$ are the (negative) Shannon entropy and KL divergence; and for any x_g, $H_{x_g}(\cdot)$ is the dilated entropy, and $D_{x_g}(\cdot\|\cdot)$ is the dilated KL divergence Hoda et al. [2010], both for the subtree rooted at x_g defined above in Definition 52.

Lemma 53 (Equivalent formulations of Φ^{Tr}-hedge)**.** *For any sequence of loss functions $\{M^t\}_{t\geq 1}$, the iterates (λ^t, m^t) in Algorithm 13 (i.e.* (7.8)-(7.10)*) are equivalent to (i.e. satisfy) the following FTRL update on H^{Tr} and OMD update on D^{Tr}:*

$$(\lambda^t, m^t) = \arg\min\nolimits_{\lambda,m} \Big[\eta \left\langle \phi(\lambda, m), \textstyle\sum_{s=1}^{t-1} M^s \right\rangle + H^{\mathsf{Tr}}(\lambda, m)\Big], \tag{7.13}$$

$$(\lambda^t, m^t) = \ \arg\min_{\lambda,m} \Big[\eta \left\langle \phi(\lambda, m), M^{t-1} \right\rangle + D^{\mathsf{Tr}}(\lambda, m \| \lambda^{t-1}, m^{t-1})\Big]. \qquad (7.14)$$

The proof of Lemma 53 follows directly by the concrete forms of (λ^t, m^t) in (7.8)-(7.10), and can be found in Appendix F.1.5.

7.1.3 Regret bound under full feedback and bandit feedback

We now present the regret bounds of Algorithm 13. We emphasize that these regret bounds are simple consequence of the generic bound for Φ-Hedge in (5.8), and their proofs do not depend on the actual implementation of Algorithm 13 developed in the preceding two subsections. We first consider the full feedback setting, where the full expected loss vector $\ell^t \in \mathbb{R}^{XA}_{\geq 0}$ is received after each episode.

Theorem 54 (Regret bound of EFCE-OMD under full feedback)**.** *Running Algorithm 13 with $\eta = \mathcal{O}(\sqrt{\|\Pi\|_1 \iota/(H^2 T)})$ achieves the following trigger regret bound*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq \mathcal{O}\big(\sqrt{H^2 \|\Pi\|_1 \iota T}\big),$$

where $\iota := \log(XA)$ is a log factor.

The proof of Theorem 54 is simply by applying (5.8) and observing that $\log(\Phi_0^{\mathsf{Tr}}) \leq \|\Pi\|_1 \log A + \log(XA)$ (see Appendix F.2.1). This theorem shows that the Φ^{Tr}-Hedge algorithm gives $\widetilde{\mathcal{O}}(\sqrt{XT})$ trigger regret bound, which matches the information-theoretic lower bound $\Omega(\sqrt{XT})$ [Zhou et al., 2020, Theorem 2] up to a $\widetilde{\mathcal{O}}(\mathrm{poly}(H))$ factor, and is slightly better than the $\widetilde{\mathcal{O}}(\sqrt{XAT})$ upper bound of [Song et al., 2022b, Corollary F.3] though their definition of EFCE-regret is slightly stricter (thus higher) than ours.

In the bandit feedback setting, the learner only observes her own rewards and infosets. In this case we replace ℓ^t in Algorithm 13 with the following loss estimator (with IX bonus γ) proposed in [Kozuno et al., 2021]:

$$\widetilde{\ell}^t_h(x_h, a_h) := \mathbf{1}\left\{(x^t_h, a^t_h) = (x_h, a_h)\right\}(1 - r^t_h)/(\mu^t_{1:h}(x_h, a_h) + \gamma). \qquad (7.15)$$

We show that EFCE-OMD achieves the following guarantee in the bandit feedback setting (proof in Appendix F.2.2). The proof follows by plugging the loss estimator $\widetilde{\ell}^t$ into (5.8) and additionally bounding concentrations (which we remark is a better strategy than using a naive bandit-based loss estimator in the corresponding NFG space).

Theorem 55 (Regret bound of EFCE-OMD under bandit feedback)**.** *Run Algorithm 13 with loss estimator* $\{\widetilde{\ell}^t\}_{t=1}^T$ (7.15)*,* $\eta = \sqrt{\|\Pi\|_1 \log A/(HXAT)}$*, and* $\gamma = \sqrt{\|\Pi\|_1 \iota/(XAT)}$*. Then we have the following trigger regret bound with probability at least* $1 - \delta$*:*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \mathcal{O}\big(\sqrt{HXA\|\Pi\|_1 \iota \cdot T}\big),$$

where $\iota = \log(3XA/\delta)$ *is a log term.*

To our best knowledge, Theorem 55 gives the first trigger regret bound against adversarial opponents and bandit feedback. This $\widetilde{\mathcal{O}}(\sqrt{XA\|\Pi\|_1 T})$ rate is $\sqrt{XA}$ worse than Theorem 54 (ignoring H and log factors), and is at most $\widetilde{\mathcal{O}}(\sqrt{X^2AT})$ using $\|\Pi\|_1 \le X$.

7.2 Balanced EFCE-OMD for bandit feedback

We now build upon the EFCE-OMD algorithm (Algorithm 13) to develop a new algorithm, *Balanced EFCE-OMD*, and show that it achieves near-optimal extensive-form trigger regret guarantee under bandit feedback. Here we discuss the two key modifications in the algorithm design, and defer the full algorithm description to Algorithm 14 .

Key modification I: "Rebalancing" the log-partition function Building on the balancing technique introduced in Ch apter 6, we start from Eq. (7.3) and (7.4) of the log partition function, and rescale the inner functions $F_{x_g a_g, x_h}$ using *balanced exploration policies* $\{\mu^{\star,h}_{g:h}(x_h, a_h)\}_{g,x_h,a_h}$, and rescale the outer function F^{Tr} by XA.

Concretely, for any matrix $M \in \mathbb{R}^{XA \times XA}$, we define the *balanced EFCE log-partition function* as

$$F^{\mathsf{Tr}}_{\mathsf{bal}}(M) := XA \log \textstyle\sum_{g, x_g, a_g} \exp\Big\{ \frac{1}{XA} \big[- \big\langle I - E_{\succeq x_g a_g}, M \big\rangle + F^{\star}_{x_g a_g, x_g}(M) \big] \Big\}, \quad (7.16)$$

where for any $x_h \succeq x_g$ (using $\mu^{\star,h}_{g:h} := \mu^{\star,h}_{g:h}(x_h, a_h)$ as shorthand, which depends on x_h but not a_h),

$$F^{\star}_{x_g a_g, x_h}(M) := \frac{1}{\mu^{\star,h}_{g:h}} \log \sum_{a_h} \exp\Big\{ \mu^{\star,h}_{g:h} \big[- M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F^{\star}_{x_g a_g, x_{h+1}}(M) \big] \Big\}. \quad (7.17)$$

Key modification II: New loss estimator under bandit feedback We use an *adaptive* family of bandit-based loss estimators $\{\widetilde{\ell}^{t, x_g a_g}\}_{x_g a_g} \subset \mathbb{R}^{XA}_{\ge 0}$, one for each $(x_g, a_g) \in \mathcal{X} \times \mathcal{A}$, defined as

$$\widetilde{\ell}^{t, x_g a_g}_h(x_h, a_h) := \frac{\mathbf{1}\{(x^t_h, a^t_h) = (x_h, a_h)\}(1 - r^t_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\{x_h \succeq x_g\})}, \quad (7.18)$$

where $\mu^t_{x_g a_g} := \mu^t_{1:g}(x_g, a_g)$ for shorthand. The main difference of (7.18) over (7.15) is in the adaptive IX bonus term on the denominator that scales with γ but is different for each $x_g a_g$. We then place each $\mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g}$ into the $x_g a_g$-th column of a matrix loss estimator $\widetilde{M}^t$, or in matrix form,

$$\widetilde{M}^t := \textstyle\sum_{g, x_g, a_g} \mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g} e^{\top}_{x_g a_g}.$$

With (7.16)-(7.18) at hand, our algorithm Balanced EFCE-OMD is defined as the negative gradient of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ evaluated at the cumulative loss estimators:

$$\phi^t = -\nabla F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\eta \textstyle\sum_{s=1}^{t-1} \widetilde{M}^s \Big), \quad \forall t \ge 1, \quad (7.19)$$

Algorithm 14 Balanced EFCE-OMD (FTRL form; equivalent OMD form in Algorithm 15)

```
Require: Learning rate $\eta$, balanced exploration policy $\{\mu^{\star,h}\}_{h\in[H]}$.
for $t = 1, 2, \dots, T$ do
    For each $x_g a_g \in \mathcal{X} \times \mathcal{A}$, from the reverse order of $x_h$, compute $m^t_{x_g a_g,h}(a_h|x_h)$ and $F^{\star,t}_{x_g a_g,x_h}$
        $m^t_{x_g a_g,h}(a_h|x_h) \propto_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\Big(-\eta\sum_{s=1}^{t-1}\widetilde{M}^s_{x_h a_h,x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^{\star,t}_{x_g a_g,x_{h+1}}\Big)\Big\},$
        $F^{\star,t}_{x_g a_g,x_h} := \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log\sum_{a_h\in\mathcal{A}} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \times \big[-\sum_{s=1}^{t}\widetilde{M}^s_{x_h a_h,x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F^{\star,t}_{x_g a_g,x_{h+1}}\big]\Big\}.$
    Compute $\lambda^{t+1}_{x_g a_g}$ as
        $\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{\frac{1}{XA}\Big(-\eta\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t-1}\widetilde{M}^s\rangle + F^{\star,t}_{x_g a_g,x_g}\Big)\Big\}.$    (7.20)
    Compute $\phi^t = \phi(\lambda^t, m^t)$, where $\phi$ is as defined in Eq. (7.7).
    Find a $\mu^t$ to be a solution of the fixed point equation $\mu^t = \phi^t\mu^t$.
    Play policy $\mu^t$, observe trajectory $(x^t_h, a^t_h, r^t_h)_{h\in[H]}$.
    Form vector loss estimator $\widetilde{\ell}^{t,x_g a_g} = \{\widetilde{\ell}^{t,x_g a_g}_h(x_h, a_h)\}_{x_h a_h}$ for each $(g, x_g a_g)$ as in Eq. (7.18).
    Compute matrix loss estimator $\widetilde{M}^t = \sum_{g,x_g,a_g} \mu^t_{x_g a_g}\widetilde{\ell}^{t,x_g a_g} e^\top_{x_g a_g}$.
```

and $\mu^t \in \Pi$ solves the fixed point equation $\phi^t\mu^t = \mu^t$. Similar as EFCE-OMD, (7.19) also admits efficient implementations in both FTRL and OMD form (cf. Algorithm 14 & 15). The corresponding (λ^t, m^t) is also equivalent to running a FTRL/OMD algorithm with respect to a *balanced* dilated entropy/KL-divergence over $\phi \in \Phi^{\mathsf{Tr}}$ (cf. Lemma 61 and Appendix 7.2.3 for details).

7.2.1 Algorithms

In this section, we present the algorithms omitted in Section 7.2. We begin with the Balanced EFCE-OMD (in FTRL form) as in Algorithm 14. This algorithm is actually equivalent to the algorithm as in Eq. (7.19) because of the following lemma, whose proof is similar to Lemma 51.

Lemma 56. *For any loss matrix $M \in \mathbb{R}_{\geq 0}^{XA \times XA}$, recall that the* balanced EFCE log-partition function *as defined in Eq. (7.16)). Let $\lambda = (\lambda_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \Delta_{XA}$ and $m = (m_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \mathcal{M}$ be*

$$\lambda_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ \frac{1}{XA} \Big(-\eta \left\langle I - E_{\succeq x_g a_g}, M \right\rangle + F^{\star}_{x_g a_g, x_g} \Big) \Big\} \tag{7.21}$$

$$m_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{ \mu^{\star, h}_{g:h}(x_h, a_h) \Big(-\eta M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^{\star}_{x_g a_g, x_{h+1}} \Big) \Big\}. \tag{7.22}$$

then we have $-\nabla F^{\mathsf{Tr}}_{\mathsf{bal}}(M) = \phi(\lambda, m)$, where ϕ is as defined in Eq. (7.7).

We also present an efficient update of (λ^{t+1}, m^{t+1}) from (λ^t, m^t), which gives the OMD form of the Balanced EFCE-OMD algorithm as in Algorithm 15. Notice the initialization of Balanced EFCE-OMD is different from EFCE-OMD (Algorithm 24) due to the presence of the balanced exploration policy. Algorithm 14 and Algorithm 15 are indeed equivalent due to the following lemma, whose proof is similar to that of Lemma 140.

Lemma 57. *Given the same sequence of M^t, Algorithm 14 and Algorithm 15 outputs the same λ^t and m^t and thus the same ϕ^t.*

7.2.2 Theoretical guarantees

We now present the theoretical guarantee of Algorithm 14 (proof in Appendix F.3).

Theorem 58. *Balanced EFCE-OMD (Algorithm 14) with $\eta = \sqrt{XA\iota / H^4 T}$ and $\gamma = 2\sqrt{XA\iota / H^2 T}$ achieves the following extensive-form trigger regret bound with probability at least $1 - \delta$:*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq \mathcal{O}\big(\sqrt{H^4 XAT\iota}\big),$$

where $\iota = \log(10XA/\delta)$ is a log term.

Algorithm 15 Balanced EFCE-OMD (OMD form; equivalent FTRL form in Algorithm 14)

```
Require: Learning rate $\eta$, balanced exploration policy $\{\mu^{\star,h}\}_{h\in[H]}$.
Initialize $\lambda^1_{x_g a_g} \propto_{x_g a_g} \exp\{(X_{\succeq x_g}/X)\log A\}$, and $m^1_{x_g a_g,h}(a_h|x_h) = 1/A$, for all $(g, x_g, a_g, h, x_h, a_h)$ with $g \le h$.
for $t = 1, 2, \dots, T$ do
    Compute $\phi^t = \phi(\lambda^t, m^t)$, where $\phi$ is as defined in Eq. (7.7).
    Find a $\mu^t$ to be a solution of the fixed point equation $\mu^t = \phi^t \mu^t$.
    Play policy $\mu^t$, observe trajectory $(x^t_h, a^t_h, r^t_h)_{h\in[H]}$.
    Form vector loss estimator $\widetilde{\ell}^{t,x_g a_g} = \{\widetilde{\ell}^{t,x_g a_g}_h(x_h, a_h)\}_{x_h a_h}$ for each $(g, x_g a_g)$ as in Eq. (7.18).
    Compute matrix loss estimator $\widetilde{M}^t = \sum_{g,x_g,a_g} \mu^t_{x_g a_g} \widetilde{\ell}^{t,x_g a_g} e^\top_{x_g a_g}$.
    For each $x_g a_g \in \mathcal{X} \times \mathcal{A}$, from the reverse order of $x_h$, compute $m^t_{x_g a_g,h}(a_h|x_h)$ and $F^{\star,t}_{x_g a_g,x_h}$
        $$\begin{aligned} m^{t+1}_{x_g a_g,h}(a_h|x_h) \propto_{a_h} & \ m^t_{x_g a_g,h}(a_h|x_h) \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \\ & \Big( -\eta \widetilde{M}^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^{\star,t}_{x_g a_g, x_{h+1}}\Big)\Big\}, \\ \widetilde{F}^{\star,t}_{x_g a_g, x_h} := & \ \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h\in\mathcal{A}} m^t_{x_g a_g,h}(a_h|x_h) \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \\ & \times \Big[ -\widetilde{M}^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} \widetilde{F}^{\star,t}_{x_g a_g, x_{h+1}}\Big]\Big\}. \end{aligned}$$
    Compute $\lambda^{t+1}_{x_g a_g}$ as
        $$\lambda^{t+1}_{x_g a_g} \propto_{x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{\frac{1}{XA}\Big( -\eta \Big\langle I - E_{\succeq x_g a_g}, \widetilde{M}^t \Big\rangle + \widetilde{F}^{\star,t}_{x_g a_g, x_g}\Big)\Big\}. \tag{7.23}$$
```

The $\widetilde{\mathcal{O}}(\sqrt{XAT})$ trigger regret asserted in Theorem 58 improves over Theorem 55 by a factor of $\sqrt{\|\Pi\|_1}$, and matches the information-theoretic lower bound up to poly(H) and log factors. As the trigger regret is lower bounded by the vanilla (external) regret, [Bai et al., 2022b, Theorem 6] implies an $\Omega(\sqrt{XAT})$ lower bound for the trigger regret as well under bandit feedback. By the online-to-batch conversion (Lemma 34), Theorem 58 also implies an $\widetilde{\mathcal{O}}(H^4XA/\varepsilon^2)$ sample complexity for learning EFCE under bandit feedback (assuming same game sizes for all m players). This improves over the best known $\widetilde{\mathcal{O}}(mH^6XA^2/\varepsilon^2)$ sample complexity in the recent work of Song et al. [2022b]. We remark though that the 1-EFR algorithm of [Song et al., 2022b] actually finds an "1-EFCE" which is slightly stronger than our EFCE defined via trigger modifications.

Overview of techniques The proof of Theorem 58 is significantly more challenging than that of Theorem 55, even though the algorithm itself is designed by appearingly simple modifications. The happens since Algorithm 14, unlike Algorithm 13, no longer necessarily corresponds to any normal-form algorithm. The technical crux of the proof is to bound the nonlinear part of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ (with respect to the losses), which we do by carefully controlling a series of second-order terms utilizing the balanced policies within $F^{\mathsf{Tr}}_{\mathsf{bal}}$ and the new adaptive IX bonus within $\{\ell^{t,x_g a_g}\}_{x_g a_g}$ (Lemma 150-153).

7.2.3 Equivalence to FTRL and OMD

Similar as Section 7.1.2, we show that the Balanced EFCE-OMD algorithm (Algorithm 14) is equivalent to FTRL with the balanced trigger dilated entropy, and OMD with the balanced dilated KL divergence, both over the (λ, m) parametrization.

We first introduce Balanced dilated entropy and balanced dilated KL divergence, and their trigger versions as below.

Definition 59 (Balanced dilated entropy and balanced dilated KL divergence)**.** The balanced dilated entropy $H^{\mathsf{bal}}_{x_g}$ rooted at x_g of subtree policy $\mu^{x_g} \in \Pi^{x_g}$ is defined as

$$H^{\mathsf{bal}}_{x_g}(\mu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \frac{\mu^{x_g}_{g:h}(x_h,a_h)}{\mu^{\star,h}_{g:h}(x_h,a_h)} \log \mu^{x_g}_{h}(a_h|x_h). \tag{7.24}$$

The balanced dilated KL divergence $D^{\mathsf{bal}}_{x_g}$ rooted at x_g between two subtree policies $\mu^{x_g}, \nu^{x_g} \in \Pi^{x_g}$ is defined as

$$D^{\mathsf{bal}}_{x_g}(\mu^{x_g} \| \nu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \frac{\mu^{x_g}_{g:h}(x_h,a_h)}{\mu^{\star,h}_{g:h}(x_h,a_h)} \log \frac{\mu^{x_g}_{h}(a_h|x_h)}{\nu^{x_g}_{h}(a_h|x_h)}. \tag{7.25}$$

Definition 60 (Balanced trigger dilated entropy and balanced trigger dilated KL divergence)**.** The balanced trigger dilated entropy function on $(\lambda, m) \in \Delta_{XA} \times \mathcal{M}$ is defined as

$$H^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m) = XA \cdot H(\lambda) + \sum_{g, x_g, a_g} \lambda_{x_g a_g} H^{\mathsf{bal}}_{x_g}(m_{x_g a_g}). \tag{7.26}$$

The balanced trigger dilated KL divergence function on $(\lambda, m), (\lambda', m') \in \Delta_{XA} \times \mathcal{M}$ is defined as

$$D^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m \| \lambda', m') = XA \cdot D_{\mathrm{KL}}(\lambda \| \lambda') + \sum_{g, x_g, a_g} \lambda_{x_g a_g} D^{\mathsf{bal}}_{x_g}(m_{x_g a_g} \| m'_{x_g a_g}). \tag{7.27}$$

The following lemma shows that the Balanced EFCE-OMD (Algorithm 14 and 15) are essentially FTRL with the balanced trigger dilated entropy, and OMD with the balanced tirgger dilated KL divergence. The proof of this lemma is similar to that of Lemma 53.

Lemma 61 (Equivalent of Balanced EFCE-OMD to OMD/FTRL on (λ, m))**.** *For any sequence of loss functions $\{\widetilde{M}^t\}_{t\geq 1}$, the algorithm as in Eq. (7.19) is equivalent to (i.e. satisfy) the following FTRL update on $H^{\mathsf{Tr}}_{\mathsf{bal}}$ and OMD update on $D^{\mathsf{Tr}}_{\mathsf{bal}}$:*

$$(\lambda^{t+1}, m^{t+1}) = \ \arg\min_{\lambda, m} \left[\eta \left\langle \phi(\lambda, m), \textstyle\sum_{s=1}^{t} \widetilde{M}^s \right\rangle + H^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m)\right], \tag{7.28}$$

$$(\lambda^{t+1}, m^{t+1}) = \ \arg\min_{\lambda, m} \left[\eta \left\langle \phi(\lambda, m), \widetilde{M}^t \right\rangle + D^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m \| \lambda^t, m^t)\right], \tag{7.29}$$

with $\phi^{t+1} = \phi(\lambda^{t+1}, m^{t+1})$.

7.3 Equivalence with existing algorithms

Interestingly, some of the algorithms we develop in this chapter is equivalent to some known algorithms which is not implemented efficiently. We present these connections in this section.

7.3.1 Equivalence of OMD and Vertex MWU

As another illustration of our framework, we now choose $\Phi = \Phi^{\mathsf{ext}} = \mathrm{conv}\{\Phi_0^{\mathsf{ext}}\}$ to be the set of *external* policy modifications, which modify any policy to some deterministic policy. In this case, the Φ^{ext}-Hedge algorithm minimizes the external regret in EFGs. In this section, we show that Φ^{ext}-Hedge, same as the vertex MWU algorithm considered in Farina et al. [2022b], is actually equivalent to the OMD with dilated entropy Hoda et al. [2010]. Let $\{\ell^t\}_{t\geq 1} \subset \mathbb{R}_{\geq 0}^{XA}$ be an arbitrary sequence of loss vectors.

Vertex MWU We use $\mathcal{V}$ to denote all the deterministic sequence-form policies, which can also be viewed as the vertex set of the policy set Π.

A simple reformulation (cf. Appendix F.4) shows that Φ^{ext}-Hedge (Algorithm 9) gives the vertex MWU algorithm considered by Farina et al. [2022b]

$$\mu^t = \textstyle\sum_{v\in\mathcal{V}} p_v^t \cdot v \qquad \text{and} \qquad p_v^t \propto_v \exp\left\{-\eta \left\langle v, \textstyle\sum_{s=1}^{t-1} \ell^s \right\rangle\right\}. \tag{7.30}$$

OMD with dilated entropy Another popular algorithm for external regret minimization is the OMD algorithm on the sequence-form policy space with the dilated entropy [Hoda et al., 2010, Kroer et al., 2015]:

$$\mu^t = \arg\min_{\mu\in\Pi} \left[\eta \left\langle \mu, \ell^{t-1} \right\rangle + D_{\emptyset}(\mu \| \mu^{t-1})\right], \tag{7.31}$$

$$D_{\emptyset}(\mu\|\nu) := \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) \log \frac{\mu_h(a_h|x_h)}{\nu_h(a_h|x_h)}. \tag{7.32}$$

Theorem 62 (Equivalence of OMD and Vertex MWU)**.** *For any sequence of loss vectors $\{\ell^t\}_{t\geq 1}$, OMD with dilated entropy is equivalent to Vertex MWU, that is,* (7.31) *and* (7.30) *give the same $\{\mu^t\}_{t\geq 1}$.*

The proof of Theorem 62 can be found in Appendix F.4.

7.3.2 Equivalence between OMD and “Kernelized MWU”

Our proof also reveals that the efficient implementation of Vertex MWU developed by Farina et al. [2022b] using the “kernel trick” is actually equivalent to the standard linear-time efficient implementation of OMD with dilated entropy. Concretely, Farina et al. [2022b] design another efficient implementation of the Vertex MWU algorithm (7.30) via the “kernel trick”, which they term as the Kernelized MWU algorithm (Algorithm 1 in [Farina et al., 2022b]). Their Algorithm 1 is an optimistic algorithm with a “prediction vector”. Here we are referring to their non-optimistic version where the prediction vectors are set to zero. As Theorem 62 shows that Vertex MWU is equivalent to standard OMD, the Kernelized MWU algorithm is also equivalent to standard OMD.

In this section, we further show that the implementation in Kernelized MWU is also “equivalent” to the standard linear-time implementation of OMD (Algorithm 25), by showing that the key intermediate quantities in both implementations are also equivalent.

Since the notation used in Farina et al. [2022b] is slightly different from ours, we first describe their key intermediate quantities using our notation. Their exponential weight $b^t \in \mathbb{R}^{XA}$ is defined by

$$b^t(x_h, a_h) = \exp\{-\eta \sum_{s=1}^{t} \ell_h^s(x_h, a_h)\}.$$

Then, their kernel function $K : \mathbb{R}^{XA} \times \mathbb{R}^{XA} \to \mathbb{R}$ is defined by

$$K_{x_g}(b, b') = \sum_{v \in \mathcal{V}^{x_g}} \sum_{(x_h, a_h) \in v} b(x_h, a_h) b'(x_h, a_h),$$

where $(x_h, a_h) \in v$ is a shorthand notation meaning that (x_h, a_h) is such that $v_{1:h}(x_h, a_h) = 1$.

We will also use $\mathbf{1} \in \mathbb{R}^{XA}$ to denote the all-ones vector in $\mathbb{R}^{XA}$. [Farina et al., 2022b, Proposition 5.3] shows that the output policy μ^t of kernelized OMWU can be written in conditional-form as

$$\mu^t(a_h|x_h) = \frac{b^{t-1}(x_h, a_h) \prod_{x_{h+1} \in \mathcal{C}(x_h a_h)} K_{x_{h+1}}(b^{t-1}, \mathbf{1})}{K_{x_h}(b^{t-1}, \mathbf{1})}.$$

The key step within Farina et al. [2022b]'s Kernelized MWU implementation is the recursive evaluation of the quantity $K_{x_h}(b^{t-1}, \mathbf{1})$ in the bottom-up order over $x_h \in \mathcal{X}$, whereas our Algorithm 25's key step is the recursive evaluation of $F^t_{x_h}$ in the bottom-up order over $x_h \in \mathcal{X}_h$ in (F.25).

The following proposition shows that these two quantities are exactly equivalent, thereby showing the equivalence of the two implementations.

Proposition 63. *We have for all $x_h \in \mathcal{X}$ and all $t \geq 1$ that*

$$K_{x_h}(b^{t-1}, \mathbf{1}) = \exp\{F^t_{x_h}\}.$$

Proof. We prove this by induction for $h = H+1, \cdots, 1$. For $h = H+1$, $K_{x_h}(b^{t-1}, \mathbf{1}) = 1$ and $F^t_{x_h} = 0$ by definition. If the claim holds for $h+1$, then by Theorem 5.2 of Farina et al. [2022b],

$$\begin{aligned} K_{x_h}(b^{t-1}, \mathbf{1}) &= \sum_{a_h} \exp\{-\eta \sum_{s=1}^{t-1} \ell^s(x_h, a_h)\} \prod_{x_{h+1} \in \mathcal{C}(x_h a_h)} K_{x_{h+1}}(b^{t-1}, \mathbf{1}) \\ &= \sum_{a_h} \exp\{-\eta \sum_{s=1}^{t-1} \ell^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F^t_{x_{h+1}}\} \\ &= \exp\{F^t_{x_h}\}. \end{aligned}$$

□

Appendix A

Proofs for Chapter 2

A.1 Proof of Proposition 5

Proof. We prove two claims separately.

For Nash ⊂ CE, let $\pi = \pi_1 \times \pi_2 \times \cdots \pi_m$ be an ε-approximate Nash equilibrium, then

$$\max_{\phi_i} V_{i,1}^{(\phi_i \diamond \pi_i)\times \pi_{-i}}(s_1) \overset{(a)}{=} \max_{\pi_i'} V_{i,1}^{\pi_i' \times \pi_{-i}}(s_1) \overset{(b)}{\leq} V_{i,1}^{\pi}(s_1) + \varepsilon.$$

Step (a) is because that π is a product policy, where the randomness of different agents are completely independent. In this case, maximizing over strategy modification ϕ_i is equivalent to maximizing over a new independent policy. Step (b) directly follows from π being an ε-approximate Nash equilibrium. By definition, this proves that π is also an ε-approximate CE.

For CE ⊂ CCE, let $\pi = \pi_1 \odot \pi_2 \odot \cdots \pi_m$ be an ε-approximate CE, then we have

$$\max_{\pi_i'} V_{i,1}^{\pi_i' \times \pi_{-i}}(s_1) \overset{(c)}{\leq} \max_{\phi_i} V_{i,1}^{(\phi_i \diamond \pi_i)\odot \pi_{-i}}(s_1) \overset{(d)}{\leq} V_{i,1}^{\pi}(s_1) + \varepsilon.$$

Step (c) is because by definition of strategy modification $\phi_i := \{\phi_{i,h} : (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i\}$, we can consider a subset of strategy modification $\phi_i' := \{\phi_{i,h}' : (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \mathcal{A}_i\}$ which modifies the policy ignoring whatever the action π_i

takes. It is not hard to see that maxmizing over the strategy modification in this subset is equivalent to maximizing over a new independent policy π_i'. Therefore, maximizing over all strategy modification is greater or equal to maximizing over π_i'. Finally, step (d) follows from π being an ε-approximate CE. By definition, this proves that π is also an ε-approximate CCE. □

A.2 Proof of Proposition 6

Proof. Let $N^\star$ be the value of Nash equilibrium for Q. Since $\pi = \mathrm{CCE}(Q, Q)$, by definition, we have:

$$\mathbb{E}_{(a,b)\sim\pi}Q(a,b) \geq \max_{a^\star}\mathbb{E}_{(a,b)\sim\pi}Q(a^\star,b) = \max_{a^\star}\mathbb{E}_{b\sim\nu}Q(a^\star,b) \geq N^\star$$
$$\mathbb{E}_{(a,b)\sim\pi}Q(a,b) \leq \min_{b^\star}\mathbb{E}_{(a,b)\sim\pi}Q(a,b^\star) = \min_{b^\star}\mathbb{E}_{a\sim\mu}Q(a,b^\star) \leq N^\star$$

This gives:

$$\max_{a^\star}\mathbb{E}_{b\sim\nu}Q(a^\star,b) = \min_{b^\star}\mathbb{E}_{a\sim\mu}Q(a,b^\star) = N^\star$$

which finishes the proof. □

A.3 Proof of the computational hardness

In this section we give the proof of the computational hardness results in Section 2.5.1, Theorem 7 and Corollary 9. Our proof is inspired by a computational hardness result for adversarial MDPs in [Yadkori et al., 2013, Section 4.2], which constructs a family of adversarial MDPs that are computationally as hard as an agnostic parity learning problem.

Section A.3.1, A.3.2, A.3.3 will be devoted to prove Theorem 7, while Corollary 9 is proved in Section A.3.4. Towards proving Theorem 7, we will:

- (Section A.3.1) Construct a Markov game.

- (Section A.3.2) Define a series of problems where a solution in problem implies another.

- (Section A.3.3) Based on the believed computational hardness of learning paries with noise (Conjecture 8), we conclude that finding the best response of non-Markov policies is computationally hard.

A.3.1 Markov game construction

We now describe a Markov game inspired the adversarial MDP in [Yadkori et al., 2013, Section 4.2]. We define a Markov game in which we have $2H$ states, $\{i_0, i_1\}_{i=2}^{H}$, 1_0 (the initial state) and $\perp$ (the terminal state)In Yadkori et al. [2013] the states are denoted by $\{i_a, i_b\}_{i=2}^{H}$ instead. Here we slightly change the notation to make it different from the notation of the actions. In each state the max-player has two actions a_0 and a_1, while the min-player has two actions b_0 and b_1. The transition kernel is deterministic and the next state for steps $h \leq H-1$ is defined in Table A.1:

State/Action	(a_0, b_0)	(a_0, b_1)	(a_1, b_0)	(a_1, b_1)
i_0	$(i+1)_0$	$(i+1)_0$	$(i+1)_0$	$(i+1)_1$
i_1	$(i+1)_1$	$(i+1)_0$	$(i+1)_1$	$(i+1)_1$

Table A.1: Transition kernel of the hard instance.

At the H-th step, i.e. states H_0 and H_1, the next state is always $\perp$ regardless of the action chosen by both players. The reward function is always 0 except at the H-th step. The reward is determined by the action of the min-player, defined by

State/Action	$(\cdot, b_0)$	$(\cdot, b_1)$
H_0	1	0
H_1	0	1

Table A.2: Reward of the hard instance.

At the beginning of every episode k, both players pick their own policies μ_k and ν_k, and execute them throughout the episode. The min-player can possibly pick her policy ν_k adaptive to all the observations in the earlier episodes. The only difference from the standard Markov game protocol is that the actions of the min-player except the last step will be revealed at the beginning of each episode, to match the setting in agnostic learning parities (Problem 2 below). Therefore we are actually considering a easier problem (for the max-player) and the lower bound naturally applies.

A.3.2 A series of computationally hard problems

We first introduce a series of problems and then show how the reduction works.

Problem 1 The max-player ε-approximates the best reponse for any general policy ν in the Markov game defined in Appendix A.3.1 with probability at least 1/2, in $\text{poly}(H, 1/\varepsilon)$ time.

Problem 2 Let $x = (x_1, \cdots, x_n)$ be a vector in $\{0{,}1\}^n$, $T \subseteq [n]$ and $0 < \alpha < 1/2$.The parity of x on T is the boolean function $\phi_T(x) = \oplus_{i \in T} x_i$. In words, $\phi_T(x)$ outputs 0 if the number of ones in the subvector $(x_i)_{i \in T}$ is even and 1 otherwise. A uniform query oracle for this problem is a randomized algorithm that returns a random uniform vector x, as well as a noisy classification $f(x)$ which is equal to $\phi_T(x)$ w.p. α and $1 - \phi_T(x)$ w.p. $1 - \alpha$. All examples returned by the oracle are independent. The learning parity with noise problem consists in designing an algorithm with access to the oracle such that,

- (**Problem 2.1**) w.p at least 1/2, find a (possibly random) function $h : \{0, 1\}^n \to \{0, 1\}$ satisy $\mathbb{E}_h P_x[h(x) \neq \phi_T(x)] \leq \varepsilon$, in $\text{poly}(n, 1/\varepsilon)$ time.

- (**Problem 2.2**) w.p at least 1/4, find $h : \{0, 1\}^n \to \{0, 1\}$ satisy $P_x[h(x) \neq \phi_T(x)] \leq \varepsilon$, in $\text{poly}(n, 1/\varepsilon)$ time.

- (**Problem 2.3**) w.p at least $1 - p$, find $h : \{0, 1\}^n \to \{0, 1\}$ satisy $P_x[h(x) \neq \phi_T(x)] \leq \varepsilon$, in $\text{poly}(n, 1/\varepsilon, 1/p)$ time.

We remark that Problem 2.3 is the formal definition of learning parity with noise [Mossel and Roch, 2005, Definition 2], which is conjectured to be computationally hard in the community (see also Conjecture 8).

Problem 2.3 reduces to Problem 2.2 Step 1: Repeatly apply algorithm for Problem 2.2 ℓ times to get $h_1, \dots, h_\ell$ such that $\min_i P_x[h_i(x) \neq \phi_T(x)] \leq \varepsilon$ with probability at least $1 - (3/4)^\ell$. This costs $\mathrm{poly}(n, \ell, 1/\varepsilon)$ time. Let $i_\star = \arg\min_i \mathrm{err}_i$ where $\mathrm{err}_i = P_x[h_i(x) \neq \phi_T(x)]$.

Step 2: Construct estimators using N additional data $(x^{(j)}, y^{(j)})_{j=1}^N$,

$$\widehat{\mathrm{err}}_i := \frac{\frac{1}{N}\sum_{j=1}^N \mathbb{I}\{h_i(x^{(j)}) \neq y^{(j)}\} - \alpha}{1 - 2\alpha}.$$

Pick $\widehat{i} = \arg\min_i \widehat{\mathrm{err}}_i$. When $N \geq \log(1/p)/\varepsilon^2$, with probability at least $1 - p/2$, we have

$$\max_i |\widehat{\mathrm{err}}_i - \mathrm{err}_i| \leq \frac{\varepsilon}{1 - 2\alpha}.$$

This means that

$$\mathrm{err}_{\widehat{i}} \leq \widehat{\mathrm{err}}_{\widehat{i}} + \frac{\varepsilon}{1 - 2\alpha} \leq \widehat{\mathrm{err}}_{i_\star} + \frac{\varepsilon}{1 - 2\alpha} \leq \mathrm{err}_{i_\star} + \frac{2\varepsilon}{1 - 2\alpha} \leq O(1)\varepsilon.$$

This step uses $\mathrm{poly}(n, N, \ell) = \mathrm{poly}(n, 1/\varepsilon, \log(1/p), \ell)$ time.

Step 3: Pick $\ell = \log(1/p)$, we are guaranteed that good events in step 1 and step 2 happen with probability $\geq 1 - p/2$ and altogether happen with probability at least $1 - p$. The total time used is $\mathrm{poly}(n, 1/\varepsilon, \log(1/p))$. Note better dependence on p than required.

Problem 2.2 reduces to Problem 2.1: If we have an algorithm that gives $\mathbb{E}_{h \sim \mathcal{D}} P_x[h(x) \neq \phi_T(x)] \leq \varepsilon$ with probability 1/2. Then if we sample $\widehat{h} \sim \mathcal{D}$, by Markov's inequality, we have with probability $\geq 1/4$ that

$$P_x[\widehat{h}(x) \neq \phi_T(x)] \leq 2\varepsilon$$

Problem 2.1 reduces to Problem 1: Consider the Markov game constructed above with $H - 1 = n$. The only missing piece we fill up here is the policy ν of the min-player, which is constructed as following. The min-player draws a sample

(x, y) from the uniform query oracle, then taking action b_0 at the step $h \leq H-1$ if $x_h = 0$ and b_1 otherwise. For the H-th step, the min-player take action b_0 if $y = 0$ and b_1 otherwise. Also notice the policy $\widehat{\mu}$ of the max-player can be descibed by a set $\widehat{T} \subseteq [H]$ where he takes action a_1 at step h if h and a_0 otherwise. As a result, the max-player receive non-zero result iff $\phi_{\widehat{T}}(x) = y$.

In the Markov game, we have $V_1^{\widehat{\mu},\nu}(s_1) = \mathbb{P}(\phi_{\widehat{T}}(x) = y)$. As a result, the optimal policy μ^* corresponds to the true parity set T. As a result,

$$(V_1^{\dagger,\nu} - V_1^{\widehat{\mu},\nu})(s_1) = \mathbb{P}_{x,y}(\phi_T(x) = y) - \mathbb{P}_{x,y}(\phi_{\widehat{T}}(x) = y) \leq \varepsilon$$

by the ε-approximation guarantee.

Also notice

$$\begin{aligned}
&\mathbb{P}_{x,y}(\phi_{\widehat{T}}(x) \neq y) - \mathbb{P}_{x,y}(\phi_T(x) \neq y) \\
=&(1-\alpha)\mathbb{P}_x(\phi_{\widehat{T}}(x) \neq \phi_T(x)) + \alpha\mathbb{P}_x(\phi_{\widehat{T}}(x) = \phi_T(x)) - \alpha \\
=&(1-2\alpha)\mathbb{P}_x(\phi_{\widehat{T}}(x) \neq \phi_T(x))
\end{aligned}$$

This implies:

$$\mathbb{P}_x(\phi_{\widehat{T}}(x) \neq \phi_T(x)) \leq \frac{\varepsilon}{1-2\alpha}$$

A.3.3 Putting them together

So far, we have proved that Solving Problem 1 implies solving Problem 2.3, where Problem 1 is the problem of learning ε-approximate best response in Markov games (the problem we are interested in), and Problem 2.3 is precisely the problem of learning parity with noise Mossel and Roch [2005]. This concludes the proof.

A.3.4 Proofs of Hardness Against Adversarial Opponents

Corollary 9 is a direct consequence of Theorem 7, as we will show now.

Proof of Corollary 9. We only need to prove a polynomial time no-regret algorithm also learns the best response in a Markov game where the min-player following non-Markov policy ν. Then the no-regret guarantee implies,

$$V_1^{\dagger,\nu}(s_1) - \frac{1}{K}\sum_{k=1}^{K} V_1^{\mu^k,\nu}(s_1) \leq \mathrm{poly}(S,H,A,B)K^{-\delta}$$

where μ_k is the policy of the max-player in the k-th episode. If we choose $\widehat{\mu}$ uniformly randomly from $\{\mu_k\}_{k=1}^{K}$, then

$$V_1^{\dagger,\nu}(s_1) - V_1^{\widehat{\mu},\nu}(s_1) \leq \mathrm{poly}(S,H,A,B)K^{-\delta}.$$

Choosing $\varepsilon = \mathrm{poly}(S,H,A,B)K^{-\delta}$, $K = \mathrm{poly}(S,H,A,B,1/\varepsilon)$ and the running time of the no-regret algorithm is still $\mathrm{poly}(S,H,A,B,1/\varepsilon)$ to learn the ε-approximate best response.

To see that the Corollary 9 remains to hold for policies that are Markovian in each episode and non-adaptive, we can take the hard instance in Theorem 7 and let ν^k denote the min-player's policy in the k-th episode. Note that each ν^k is Markovian and non-adaptive on the observations in previous episodes. If there is a polynomial time no-regret algorithm against such $\left\{\nu^k\right\}$, then by the online-to-batch conversion similar as the above, the mixture of $\{\mu_k\}_{k=1}^{K}$ learns a best response against ν in polynomial time.

□

A.4 Proof of the statistical hardness

In this section we prove the statistical lower bounds in Section 2.5.2, Theorem 10 and Lemma 11.

The lower bound builds on the following lower bound for adversarial MDPs where both the transition and the reward function of each episode are chosen adversarially. We state it here as a formal version of Lemma 11.

Lemma 64 (Lower bound for adversarial MDPs)**.** *For any horizon $H \geq 2$ and $K \geq 1$, there exists a family of MDPs $\mathcal{M}$ with horizon H, state space $\{S_h\}_{h \leq H}$ with $|S_h| \leq 2$, action space $\{A_h\}_{h \leq H}$ with $|A_h| \leq 2$, and reward $r_h \in [0,1]$ such that the following is true: for any algorithm that deploys policy μ^k in episode k, we have*

$$\sup_{M_1, \cdots, M_K \in \mathcal{M}} \sup_{\mu} \sum_{k=1}^{K} \left(V_{M_k}^{\mu}(s_0) - \mathbb{E}_{\mu_k} V_{M_k}^{\mu^k}(s_0) \right) \geq \Omega(\min \left\{ \sqrt{2^H K}, K \right\}),$$

where $V_{M_k}^*$ refers to the optimal value function of MDP M_k.

As we shall see in our proof of Lemma 64, the optimal policies for M_k are the same, so Lemma 64 indeed implies a lower bound on the regret defined against the best stationary policy in hindsight.

Proof. Our construction is inspired by the "combination lock" MDP [Du et al., 2019]. Let us redefine the horizon length as $H+1$ (so that $H \geq 1$) and let h start from 0. We now define our family of MDPs.

Definition 65 (MDP $M_{X,Y,\varepsilon}$)**.** For any pair of bit strings $X = (x_1, \ldots, x_H) \in \{0,1\}^H$, $Y = (y_1, \ldots, y_H) \in \{0,1\}^H$ and any $\varepsilon \in (0,1)$, the MDP $M_{X,Y,\varepsilon}$ is defined as follows.

1. The state space is $S_0 = \{s_0\}$ and $S_h = \{s_{0,h}, s_{1,h}\}$ for all $1 \leq h \leq H$. The MDP starts at s_0 deterministically and terminates at $s_{0,H}$ or $s_{1,H}$.

2. The action space is $A_h = \{0,1\}$ for all $0 \leq h \leq H$.

3. The transition is defined as follows:

   - s_0 transitions to $s_{0,1}$ or $s_{1,1}$ with probability at least $1/2$ each, regardless of the action taken.
   - For any $1 \leq h \leq H-1$, $s_{y_h,h}$ transitions to $s_{y_{h+1},h+1}$ deterministically if $a_h = x_h \oplus y_h$ ("correct state" in combination lock), and transitions to $s_{1-y_{h+1},h+1}$ deterministically if $a_h = 1 - x_h \oplus y_h$.
   - For any $1 \leq h \leq H-1$, $s_{1-y_h,h}$ transitions to $s_{1-y_{h+1},h+1}$ deterministically regardless of the action taken ("wrong state" in combination lock).

4. The reward is $r_h \equiv 0$ for all $0 \le h \le H-1$. At step H, we have

   - $r_H(s_{y_H,H}) \sim \mathsf{Ber}(1/2+\varepsilon)$,
   - $r_H(s_{1-y_H,H}) \sim \mathsf{Ber}(1/2-\varepsilon)$.

A visualization for the MDP specified by X, Y and ε is shown in Figure A-1.

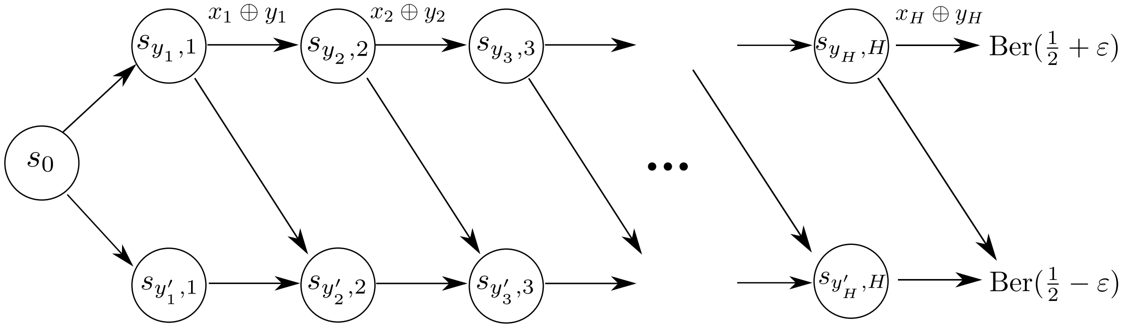


Figure A-1: $M(X,Y)$: "Combination lock" MDP specified by X and Y. For $y \in \{0,1\}$, y' stands for $1-y$.

It is straightforward to see that the optimal value function of this MDP is $1/2(1/2+\varepsilon)+1/2(1/2-\varepsilon)=1/2$, and the only way to achieve higher reward than $1/2-\varepsilon$ is by following the path of "good states": $(s_0, s_{y_1,1}, \cdots, s_{y_h,h}, \cdots, s_{y_H,H})$. The corresponding optimal policy is $\pi^*(s_{w,h}) = w \oplus x_h$, which is independent of Y.

Random sequence of MDPs is as hard as a 2^H-armed bandit. We now consider any fixed (but unknown) $X \in \{0,1\}^H$ and draw K independent samples $Y_k \sim \mathsf{Unif}(\{0,1\}^H)$ for $1 \le k \le K$. We argue that if we provide $M_k := M_{X,Y_k,\varepsilon}$ in episode k (with some appropriate choice of ε), then the problem is as hard as a 2^H-armed bandit problem with (minimum) suboptimality gap ε, and thus must have the desired regret lower bound.

Our first claim is that, on average over Y_k, the trajectory seen by the algorithm is equivalent (equal in distribution) to the following "completely random" MDP: each state $s_{\{0,1\},h}$ transitions to $s_{\{0,1\},h+1}$ with probability at least $1/2$ regardless of the actions taken; and the reward is $r_H \sim \mathsf{Ber}(1/2)$ if $A = X \oplus Y$ and $r_H \sim \mathsf{Ber}(1/2-\varepsilon)$ if $A \neq X \oplus Y$, where $A = \{a_1, \ldots, a_h\}$ are the actions taken in steps 1 through

H. Indeed, consider the transition starting from $s_{y_h,h}$. Since $y_{h+1} \sim \mathsf{Ber}(1/2)$, the transition probability to $s_{0,h+1}$ and $s_{1,h+1}$ must be $1/2$ each, regardless of the action taken. The claim about the reward follows from the definition of the MDP.

We now construct a bandit instance, and show that solving this bandit problem can be reduced to online learning in the sequence of MDPs above. The bandit instance has 2^H arms indexed by $\{0,1\}^H$. The arm indexed by X gives reward $\mathsf{Ber}(1/2)$, and otherwise the reward is $\mathsf{Ber}(1/2-\varepsilon)$. Now, for any algorithm solving the adversarial MDP problem, consider the following induced algorithm for the bandit problem.

Algorithm 16 Reducing bandits to adversarial MDPs

```
for $k = 1, \dots, K$ do
    Sample $Y \sim \mathsf{Unif}(\{0,1\}^H)$.
    Simulate the adversarial MDP algorithm by showing the trajectory $(s_0, s_{y_1,1}, \dots, s_{y_H,H})$.
    Denote the action sequence by $A = (a_1, \dots, a_H)$.
    Play $A \oplus Y$ in the bandit environment.
    Show the received bandit reward to the adversarial MDP algorithm as the last step reward.
```

We now argue that the interaction seen by the adversarial MDP algorithm is identical in distribution to the sequence $M_{X,Y_k,\varepsilon}$. The trajectory is drawn from a uniform distribution, which is the same as that generated by $M_{X,Y_k,\varepsilon}$. The reward is high, i.e. $\mathsf{Ber}(1/2)$, if and only if $A \oplus Y = X$, which is equivalent to $A = X \oplus Y$. This is also the case in the adversarial MDP problem, since playing the action sequence $X \oplus Y$ corresponds to playing the optimal policy $\pi^*(s_{y_h,h}) = x_h \oplus y_h$.

Therefore, the regret achieved by the induced algorithm in the bandit environment would be equal (in distribution) to the regret achieved by this algorithm in the adversarial MDP environment. Applying classical lower bounds on stochastic bandits [Lattimore and Szepesvári, 2020, Chapter 15] (which corresponds to taking $\varepsilon = \varepsilon_{H,K} := \min\left\{\sqrt{2^H/K}, 1/4\right\}$), we obtain

$$\sup_{X \in \{0,1\}^H} \mathbb{E}_{Y_1,\dots,Y_k \sim \mathsf{Unif}(\{0,1\}^H)} \left[\sum_{k=1}^{K} \left(V^*_{M_{X,Y_k,\varepsilon_{H,K}}}(s_0) - \mathbb{E}_{\mu^k} V^{\mu^k}_{M_{X,Y_k,\varepsilon_{H,K}}}(s_0) \right) \right] \\ \geq \Omega(\min\left\{\sqrt{2^H K}, K\right\}),$$

where $\mathbb{E}_{\mu^k}$ denotes the randomness in the algorithm execution (which includes the randomness of the realized transitions and rewards that were used by the algorithm to determine μ^k). Note that for the MDP $M_{X,Y_k,\varepsilon_H,T}$, the optimal policy is dictated by X and independent of Y_k (hence independent of k). Thus, the previous lower bound can rewritten as a comparison with the best policy in hindsight:

$$\sup_{X\in\{0,1\}^H} \sup_{\mu} \mathbb{E}_{Y_1,\dots,Y_k\sim\mathsf{Unif}(\{0,1\}^H)} \left[\sum_{k=1}^{K} \left(V^{\mu}_{M_{X,Y_k,\varepsilon_{H,K}}}(s_0) - \mathbb{E}_{\mu^k} V^{\mu^k}_{M_{X,Y_k,\varepsilon_{H,K}}}(s_0)\right)\right] \geq \Omega(\min\left\{\sqrt{2^H K}, K\right\}).$$

The adversarial MDP problem is as hard as the above random sequence of MDPs. Define $\mathcal{M} := \left\{M_{X,Y,\varepsilon_{H,K}} : X, Y \in \{0,1\}^H\right\}$. As the minimax regret is lower bounded by the average regret over any prior distribution of MDPs, the above lower bound implies the following minimax lower bound

$$\sup_{M_k\in\mathcal{M}} \sup_{\mu} \left[\sum_{k=1}^{K} \left(V^{\mu}_{M_k}(s_0) - \mathbb{E}_{\mu^k} V^{\mu^k}_{M_k}(s_0)\right)\right] \geq \Omega(\min\left\{\sqrt{2^H K}, K\right\})$$

for any adversarial MDP algorithm. □

Proof of Theorem 1. With Lemma 64 in hand, we are in a position to prove the main theorem.

Our proof follows by defining a two-player Markov game and a set of min-player policies $\left\{\nu^k\right\}$ such that the transitions and rewards seen by the max-player are exactly equivalent to the MDP $M_{X,Y_k,\varepsilon_{H,K}}$ constructed in Lemma 64. Indeed, we augment the MDP $M_{X,Y_k,\varepsilon_{H,K}}$ with a set of min-player actions $\mathcal{B}_h = \{1,2,3,4\}$, and redefine the transition such that from any $s_{i,h}$ where $i \in \{0,1\}$ and $1 \leq h \leq H-1$, the Markov game transitions according to Table A.3.

a/b	1	2	3	4
0	$s_{i,h+1}$	$s_{1-i,h+1}$	$s_{i,h+1}$	$s_{1-i,h+1}$
1	$s_{i,h+1}$	$s_{1-i,h+1}$	$s_{1-i,h+1}$	$s_{i,h+1}$

Table A.3: transition function of the state $s_{i,h}$ for the hard instance of Markov games.

Such an action set $\mathcal{B}_h$ is powerful enough to reproduce all the possible transitions in the original single-player MDP. We then define ν^k as the policy such that the transition follows exactly M_{X,Y_k}. The reward function is determined only by states and thus remains the same. Therefore, Lemma 64 implies the following one-sided regret bound for the max-player:

$$\sup_{\nu^k} \sup_{\mu} \sum_{k=1}^{K} \left(V^{\mu,\nu^k}(s_0) - \mathbb{E}_{\mu^k} V^{\mu^k,\nu^k}(s_0) \right) \geq \Omega(\min \left\{ \sqrt{2^H K}, K \right\}),$$

which is the desired result. □

Appendix B

Proofs for Chapter 3

B.1 Proof for Section 3.2 – Optimistic Nash Value Iteration

B.1.1 Proof of Theorem 16

We denote V^k, Q^k, π^k, μ^k and ν^k [1] for values and policies at the *beginning* of the k-th episode. In particular, $N_h^k(s,a,b)$ is the number we have visited the state-action tuple (s,a,b) at the h-th step before the k-th episode. $N_h^k(s,a,b,s')$ is defined by the same token. Using this notation, we can further define the empirical transition by $\widehat{\mathbb{P}}_h^k(s'|s,a,b) := N_h^k(s,a,b,s')/N_h^k(s,a,b)$. If $N_h^k(s,a,b)=0$, we set $\widehat{\mathbb{P}}_h^k(s'|s,a,b) = 1/S$.

As a result, the bonus terms can be written as

$$\beta_h^k(s,a,b) := C\left(\sqrt{\frac{\iota H^2}{\max\{N_h^k(s,a,b),1\}}} + \frac{H^2 S\iota}{\max\{N_h^k(s,a,b),1\}}\right) \tag{B.1}$$

$$\gamma_h^k(s,a,b) := \frac{C}{H}\widehat{\mathbb{P}}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b) \tag{B.2}$$

for some large absolute constant $C > 0$.

[1] recall that (μ_h^k, ν_h^k) are the marginal distributions of π_h^k.

Lemma 66. *Let c_1 be some large absolute constant. Define event E_0 to be: for all h, s, a, b, s' and $k \in [K]$,*

$$\begin{cases} |[(\widehat{\mathbb{P}}_h^k - \Pr_h) V_{h+1}^\star](s,a,b)| \le c_1 \sqrt{\dfrac{H^2 \iota}{\max\{N_h^k(s,a,b), 1\}}}, \\ |(\widehat{\mathbb{P}}_h^k - \Pr_h)(s' \mid s,a,b)| \le c_1 \left(\sqrt{\dfrac{\min\{\Pr_h(s' \mid s,a,b), \widehat{\mathbb{P}}_h^k(s' \mid s,a,b)\}\iota}{\max\{N_h^k(s,a,b),1\}}} + \dfrac{\iota}{\max\{N_h^k(s,a,b),1\}} \right). \end{cases}$$

We have $\Pr(E_1) \ge 1 - p$.

Proof. The proof is standard and folklore: apply standard concentration inequalities and then take a union bound. For completeness, we provide the proof of the second one here.

Consider a fixed (s, a, b, h) tuple.

Let's consider the following equivalent random process: (a) before the agent starts, the environment samples $\{s^{(1)}, s^{(2)}, \ldots, s^{(K)}\}$ independently from $\Pr_h(\cdot \mid s,a,b)$; (b) during the interaction between the agent and environment, the i^{th} time the agent reaches (s,a,b,h), the environment will make the agent transit to $s^{(i)}$. Note that the randomness induced by this interaction procedure is exactly the same as the original one, which means the probability of any event in this context is the same as in the original problem. Therefore, it suffices to prove the target concentration inequality in this 'easy' context. Denote by $\widehat{\mathbb{P}}_h^{(t)}(\cdot \mid s,a,b)$ the empirical estimate of $\Pr_h(\cdot \mid s,a,b)$ calculated using $\{s^{(1)}, s^{(2)}, \ldots, s^{(t)}\}$. For a fixed t and s', by applying the Bernstein inequality and its empirical version, we have with probability at least $1 - p/S^2ABT$,

$$|(\Pr_h - \widehat{\mathbb{P}}_h^{(t)})(s' \mid s,a,b)| \le \mathcal{O}\left(\sqrt{\frac{\min\{\Pr_h(s' \mid s,a,b), \widehat{\mathbb{P}}_h^{(t)}(s' \mid s,a,b)\}\iota}{t}} + \frac{\iota}{t} \right).$$

Now we can take a union bound over all s, a, b, h, s' and $t \in [K]$, and obtain that with probability at least $1-p$, for all s, a, b, h, s' and $t \in [K]$,

$$|(\Pr_h - \widehat{\mathbb{P}}_h^{(t)})(s' \mid s, a, b)| \leq \mathcal{O}\left(\sqrt{\frac{\min\{\Pr_h(s' \mid s, a, b), \widehat{\mathbb{P}}_h^{(t)}(s' \mid s, a, b)\}\iota}{t}} + \frac{\iota}{t}\right).$$

Note that the agent can reach each (s, a, b, h) for at most K times, this directly implies that the third inequality also holds with probability at least $1-p$. □

We begin with an auxiliary lemma bounding the lower-order term.

Lemma 67. *Suppose event E_0 holds, then there exists absolute constant c_2 such that: if function $g(s)$ satisfies $|g|(s) \leq (\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s)$ for all s, then*

$$\begin{aligned}
&|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)g(s, a, b)| \\
\leq& c_2\left(\frac{1}{H}\min\{\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s, a, b), \mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s, a, b)\} + \frac{H^2 S\iota}{\max\{N_h^k(s, a, b), 1\}}\right).
\end{aligned}$$

Proof. By triangle inequality,

$$\begin{aligned}
&|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)g(s, a, b)| \\
\leq& \sum_{s'} |(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(s'|s, a, b)||g|(s') \\
\leq& \sum_{s'} |(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(s'|s, a, b)|(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s') \\
\overset{(i)}{\leq}& \mathcal{O}\left(\sum_{s'}(\sqrt{\frac{\iota\widehat{\mathbb{P}}_h^k(s'|s, a, b)}{\max\{N_h^k(s, a, b), 1\}}} + \frac{\iota}{\max\{N_h^k(s, a, b), 1\}})(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s')\right) \\
\overset{(ii)}{\leq}& \mathcal{O}\left(\sum_{s'}(\frac{\widehat{\mathbb{P}}_h^k(s'|s, a, b)}{H} + \frac{H\iota}{\max\{N_h^k(s, a, b), 1\}})(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s')\right) \\
\leq& \mathcal{O}\left(\frac{\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s, a, b)}{H} + \frac{H^2 S\iota}{\max\{N_h^k(s, a, b), 1\}}\right),
\end{aligned}$$

where (i) is by the second inequality in event E_0 and (ii) is by AM-GM inequality.

This proves the empirical version. Similarly, we can show

$$|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)g(s,a,b)| \le \mathcal{O}\left(\frac{\Pr_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b)}{H} + \frac{H^2 S\iota}{\max\{N_h^k(s,a,b),1\}}\right),$$

Combining the two bounds completes the proof. □

Now we can prove the upper and lower bounds are indeed upper and lower bounds of the best reponses.

Lemma 68. *Suppose event E_0 holds. Then for all h, s, a, b and $k \in [K]$, we have*

$$\begin{cases} \overline{Q}_h^k(s,a,b) \ge Q_h^{\dagger,\nu^k}(s,a,b) \ge Q_h^{\mu^k,\dagger}(s,a,b) \ge \underline{Q}_h^k(s,a,b), \\ \overline{V}_h^k(s) \ge V_h^{\dagger,\nu^k}(s) \ge V_h^{\mu^k,\dagger}(s) \ge \underline{V}_h^k(s). \end{cases} \tag{B.3}$$

Proof. The proof is by backward induction. Suppose the bounds hold for the Q-values in the $(h+1)^{\text{th}}$ step, we now establish the bounds for the V-values in the $(h+1)^{\text{th}}$ step and Q-values in the h^{th}-step. For any state s:

$$\begin{aligned} \overline{V}_{h+1}^k(s) &= \mathbb{D}_{\pi_{h+1}^k}\overline{Q}_{h+1}^k(s) \\ &\ge \max_{\mu} \mathbb{D}_{\mu\times\nu_{h+1}^k}\overline{Q}_{h+1}^k(s) \\ &\ge \max_{\mu} \mathbb{D}_{\mu\times\nu_{h+1}^k} Q_{h+1}^{\dagger,\nu^k}(s) = V_{h+1}^{\dagger,\nu^k}(s). \end{aligned} \tag{B.4}$$

Similarly, we can show $\underline{V}_{h+1}^k(s) \le V_{h+1}^{\mu^k,\dagger}(s)$. Therefore, we have: for all s,

$$\overline{V}_{h+1}^k(s) \ge V_{h+1}^{\dagger,\nu^k}(s) \ge V_{h+1}^{\star}(s) \ge V_{h+1}^{\mu^k,\dagger}(s) \ge \underline{V}_{h+1}^k(s).$$

Now consider an arbitrary triple (s, a, b) in the h^{th} step. We have

$$\begin{aligned}
&(\overline{Q}_h^k - Q_h^{\dagger,\nu^k})(s,a,b) \\
\geq& \min\left\{(\widehat{\mathbb{P}}_h^k \overline{V}_{h+1}^k - \mathbb{P}_h V_{h+1}^{\dagger,\nu^k} + \beta_h^k + \gamma_h^k)(s,a,b), 0\right\} \\
\geq& \min\left\{(\widehat{\mathbb{P}}_h^k V_{h+1}^{\dagger,\nu^k} - \mathbb{P}_h V_{h+1}^{\dagger,\nu^k} + \beta_h^k + \gamma_h^k)(s,a,b), 0\right\} \\
=& \min\left\{\underbrace{(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(V_{h+1}^{\dagger,\nu^k} - V_{h+1}^\star)(s,a,b)}_{(A)} + \underbrace{(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star(s,a,b)}_{(B)} + (\beta_h^k + \gamma_h^k)(s,a,b), 0\right\}.
\end{aligned} \tag{B.5}$$

Invoking Lemma 67 with $g = V_{h+1}^{\dagger,\nu^k} - V_{h+1}^\star$,

$$|(A)| \leq \mathcal{O}\left(\frac{\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b)}{H} + \frac{H^2 S\iota}{\max\{N_h^k(s,a,b), 1\}}\right).$$

By the first inequality in event E_0,

$$|(B)| \leq \mathcal{O}\left(\sqrt{\frac{H^2\iota}{\max\{N_h^k(s,a,b), 1\}}}\right).$$

Plugging the two inequalities above back into (B.5) and recalling the definition of β_h^k and γ_h^k, we obtain $\overline{Q}_h^k(s,a,b) \geq Q_h^{\dagger,\nu^k}(s,a,b)$. Similarly, we can show $\underline{Q}_h^k(s,a,b) \leq Q_h^{\mu^k,\dagger}(s,a,b)$. □

Finally we come to the proof of Theorem 16.

Proof of Theorem 16. Suppose event E_0 holds. We first upper bound the regret. By Lemma 68, the regret can be upper bounded by

$$\sum_k (V_1^{\dagger,\nu^k}(s_1^k) - V_1^{\mu^k,\dagger}(s_1^k)) \leq \sum_k (\overline{V}_1^k(s_1^k) - \underline{V}_1^k(s_1^k)).$$

For brevity's sake, we define the following notations:

$$\begin{cases} \Delta_h^k := (\overline{V}_h^k - \underline{V}_h^k)(s_h^k), \\ \zeta_h^k := \Delta_h^k - (\overline{Q}_h^k - \underline{Q}_h^k)(s_h^k, a_h^k, b_h^k), \\ \xi_h^k := \mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s_h^k, a_h^k, b_h^k) - \Delta_{h+1}^k. \end{cases} \tag{B.6}$$

Let $\mathcal{F}_h^k$ be the σ-field generated by the following random variables:

$$\{(s_i^j, a_i^j, b_i^j, r_i^j)\}_{(i,j)\in[H]\times[k-1]} \bigcup \{(s_i^k, a_i^k, b_i^k, r_i^k)\}_{i\in[h-1]} \bigcup \{s_h^k\}.$$

It's easy to check ζ_h^k and ξ_h^k are martingale differences with respect to $\mathcal{F}_h^k$. With a slight abuse of notation, we use β_h^k to refer to $\beta_h^k(s_h^k, a_h^k, b_h^k)$ and N_h^k to refer to $N_h^k(s_h^k, a_h^k, b_h^k)$ in the following proof.

We have

$$\begin{aligned} \Delta_h^k =& \zeta_h^k + \left(\overline{Q}_h^k - \underline{Q}_h^k\right)\left(s_h^k, a_h^k, b_h^k\right) \\ \le& \zeta_h^k + 2\beta_h^k + 2\gamma_h^k + \widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)\left(s_h^k, a_h^k, b_h^k\right) \\ \overset{(i)}{\le}& \zeta_h^k + 2\beta_h^k + 2\gamma_h^k + \mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)\left(s_h^k, a_h^k, b_h^k\right) \\ & \qquad + c_2\left(\frac{\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s_h^k, a_h^k, b_h^k)}{H} + \frac{H^2 S\iota}{\max\{N_h^k, 1\}}\right) \\ \overset{(ii)}{\le}& \zeta_h^k + 2\beta_h^k + \mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)\left(s_h^k, a_h^k, b_h^k\right) \\ & \qquad + 2c_2C\left(\frac{\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s_h^k, a_h^k, b_h^k)}{H} + \frac{H^2 S\iota}{\max\{N_h^k, 1\}}\right) \\ \le& \zeta_h^k + \left(1 + \frac{2c_2C}{H}\right)\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)\left(s_h^k, a_h^k, b_h^k\right) \\ & \qquad + 4c_2C\left(\sqrt{\frac{\iota H^2}{\max\{N_h^k, 1\}}} + \frac{H^2 S\iota}{\max\{N_h^k, 1\}}\right) \\ =& \zeta_h^k + \left(1 + \frac{2c_2C}{H}\right)\xi_h^k + \left(1 + \frac{2c_2C}{H}\right)\Delta_{h+1}^k + 4c_2C\left(\sqrt{\frac{\iota H^2}{\max\{N_h^k, 1\}}} + \frac{H^2 S\iota}{\max\{N_h^k, 1\}}\right) \end{aligned}$$

where (i) and (ii) follow from Lemma 67.

Define $c_3 := 1 + 2c_2C$ and $\kappa := 1 + c_3/H$. Recursing this argument for $h \in [H]$ and summing over k,

$$\sum_{k=1}^{K} \Delta_1^k \leq \sum_{k=1}^{K}\sum_{h=1}^{H} \left[\kappa^{h-1}\zeta_h^k + \kappa^h \xi_h^k + \mathcal{O}\left(\sqrt{\frac{\iota H^2}{\max\{N_h^k, 1\}}} + \frac{H^2 S\iota}{\max\{N_h^k, 1\}}\right)\right].$$

By Azuma-Hoeffding inequality, with probability at least $1 - p$,

$$\begin{cases} \sum_{k=1}^{K}\sum_{h=1}^{H} \kappa^{h-1}\zeta_h^k \leq \mathcal{O}\Big(H\sqrt{HK\iota}\Big) = \mathcal{O}\Big(\sqrt{H^2T\iota}\Big), \\ \sum_{k=1}^{K}\sum_{h=1}^{H} \kappa^{h}\xi_h^k \leq \mathcal{O}\Big(H\sqrt{HK\iota}\Big) = \mathcal{O}\Big(\sqrt{H^2T\iota}\Big). \end{cases} \tag{B.7}$$

By pigeon-hole argument,

$$\begin{aligned} \sum_{k=1}^{K}\sum_{h=1}^{H} \frac{1}{\sqrt{\max\{N_h^k, 1\}}} \leq & \sum_{s,a,b,h:\ N_h^K(s,a,b)>0} \ \sum_{n=1}^{N_h^K(s,a,b)} \frac{1}{\sqrt{n}} + HSAB \\ \leq & \mathcal{O}\Big(\sqrt{HSABT} + HSAB\Big), \end{aligned}$$

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \frac{1}{\max\{N_h^k, 1\}} \leq \sum_{s,a,b,h:\ N_h^K(s,a,b)>0} \ \sum_{n=1}^{N_h^K(s,a,b)} \frac{1}{n} + HSAB \leq \mathcal{O}(HSAB\iota).$$

Put everything together, with probability at least $1 - 2p$ (one p comes from $\Pr(E_0) \geq 1 - p$ and the other is for equation (B.7)),

$$\sum_{k=1}^{K} (V_1^{\dagger,\nu^k}(s_1^k) - V_1^{\mu^k,\dagger}(s_1^k)) \leq \mathcal{O}\Big(\sqrt{H^3SABT\iota} + H^3S^2AB\iota^2\Big)$$

For the PAC guarantee, recall that we choose $\pi^{\text{out}} = \pi^{k^\star}$ such that

$$k^\star := \arg\min_k \left(\overline{V}_1^k - \underline{V}_1^k\right)(s_1).$$

As a result,

$$(V_1^{\dagger,\nu^{k^\star}} - V_1^{\mu^{k^\star},\dagger})(s_1) \le (\overline{V}_1^{k^\star} - \underline{V}_1^{k^\star})(s_1) \le \frac{1}{K}\mathcal{O}\Big(\sqrt{H^3SABT\iota} + H^3S^2AB\iota^2\Big),$$

which concludes the proof. □

B.1.2 Proof of Theorem 17

We use the same notation as in Appendix B.1.1 except the form of bonus. Besides, we define the empirical variance operator

$$\widehat{\mathbb{V}}_h^k V(s,a,b) := \mathrm{Var}_{s'\sim\widehat{\mathbb{P}}_h^k(\cdot|s,a,b)} V(s')$$

and the true (population) variance operator

$$\mathbb{V}_h V(s,a,b) := \mathrm{Var}_{s'\sim\mathbb{P}_h(\cdot|s,a,b)} V(s')$$

for any function $V \in \Delta^S$. If $N_h^k(s,a,b) = 0$, we simply set $\widehat{\mathbb{V}}_h^k V(s,a,b) := H^2$ regardless of the choice of V.

As a result, the bonus terms can be written as

$$\beta_h^k(s,a,b) := C\left(\sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} + \frac{H^2S\iota}{\max\{N_h^k(s,a,b),1\}}\right) \tag{B.8}$$

for some absolute constant $C > 0$.

Lemma 69. *Let c_1 be some large absolute constant. Define event E_1 to be: for all*

h, s, a, b, s' and $k \in [K]$,

$$\begin{cases} |[(\widehat{\mathbb{P}}_h^k - \Pr_h)V_{h+1}^\star](s,a,b)| \le c_1 \left(\sqrt{\dfrac{\widehat{\mathbb{V}}_h^k V_{h+1}^\star(s,a,b)\iota}{\max\{N_h^k(s,a,b),1\}}} + \dfrac{H\iota}{\max\{N_h^k(s,a,b),1\}} \right), \\ |(\widehat{\mathbb{P}}_h^k - \Pr_h)(s' \mid s,a,b)| \le c_1 \left(\sqrt{\dfrac{\min\{\Pr_h(s' \mid s,a,b), \widehat{\mathbb{P}}_h^k(s' \mid s,a,b)\}\iota}{\max\{N_h^k(s,a,b),1\}}} + \dfrac{\iota}{\max\{N_h^k(s,a,b),1\}} \right), \\ \|(\widehat{\mathbb{P}}_h^k - \Pr_h)(\cdot \mid s,a,b)\|_1 \le c_1 \sqrt{\dfrac{S\iota}{\max\{N_h^k(s,a,b),1\}}}. \end{cases}$$

We have $\Pr(E_1) \ge 1 - p$.

The proof of Lemma 69 is highly similar to that of Lemma 66. Specifically, the first two can be proved by following basically the same argument in Lemma 66; the third one is standard (e.g., equation (12) in Azar et al. [2017]). We omit the proof here.

Since the proof of Lemma 67 does not depend on the form of the bonus, it can also be applied in this section. As in Appendix B.1.1, we will prove the upper and lower bounds are indeed upper and lower bounds of the best reponses.

Lemma 70. *Suppose event E_1 holds. Then for all h, s, a, b and $k \in [K]$, we have*

$$\begin{cases} \overline{Q}_h^k(s,a,b) \ge Q_h^{\dagger,\nu^k}(s,a,b) \ge Q_h^{\mu^k,\dagger}(s,a,b) \ge \underline{Q}_h^k(s,a,b), \\ \overline{V}_h^k(s) \ge V_h^{\dagger,\nu^k}(s) \ge V_h^{\mu^k,\dagger}(s) \ge \underline{V}_h^k(s). \end{cases} \tag{B.9}$$

Proof. The proof is by backward induction and very similar to that of Lemma 68. Suppose the bounds hold for the Q-values in the $(h+1)^{\text{th}}$ step, we now establish the bounds for the V-values in the $(h+1)^{\text{th}}$ step and Q-values in the h^{th}-step.

The proof for the V-values is the same as (B.4).

For the Q-values, the decomposition (B.5) still holds and (A) is bounded using Lemma 67 as before. The only difference is that we need to bound (B) more carefully.

First, by the first inequality in event E_1,

$$|(B)| \le \mathcal{O}\left(\sqrt{\frac{\widehat{\mathbb{V}}_h^k V_{h+1}^{\star}(s,a,b)\iota}{\max\{N_h^k(s,a,b),1\}}} + \frac{H\iota}{\max\{N_h^k(s,a,b),1\}}\right).$$

By the relation of V-values in the $(h+1)^{\text{th}}$ step,

$$\begin{aligned}
&|[\widehat{\mathbb{V}}_h^k(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2] - \widehat{\mathbb{V}}_h^k V_{h+1}^{\star}|(s,a,b) \\
\le& |[\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2]^2 - (\widehat{\mathbb{P}}_h^k V_{h+1}^{\star})^2|(s,a,b) \\
&\qquad + |\widehat{\mathbb{P}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2]^2 - \widehat{\mathbb{P}}_h^k(V_{h+1}^{\star})^2|(s,a,b) \\
\le& 4H\widehat{\mathbb{P}}_h^k|(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2 - V_{h+1}^{\star}|(s,a,b) \\
\le& 4H\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b),
\end{aligned} \tag{B.10}$$

which implies

$$\begin{aligned}
&\sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k V_{h+1}^{\star}(s,a,b)}{\max\{N_h^k(s,a,b),1\}}} \\
\le& \sqrt{\frac{\iota[\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2] + 4H\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} \\
\le& \sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} + \sqrt{\frac{4\iota H\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} \\
\overset{(i)}{\le}& \sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} + \frac{\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)}{H} + \frac{4H^2\iota}{\max\{N_h^k(s,a,b),1\}},
\end{aligned} \tag{B.11}$$

where (i) is by AM-GM inequality.

Plugging the above inequalities back into (B.5) and recalling the definition of β_h^k and γ_h^k completes the proof. □

We need one more lemma to control the error of the empirical variance estimator:

Lemma 71. *Suppose event E_1 holds. Then for all h, s, a, b and $k \in [K]$, we have*

$$|\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2] - \mathbb{V}_h V_{h+1}^{\pi^k}|(s,a,b)$$

$$\le 4H \Pr_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b) + \mathcal{O}\left(1+\frac{H^4 S\iota}{\max\{N_h^k(s,a,b),1\}}\right).$$

Proof. By Lemma 70, we have $\overline{V}_h^k(s) \ge V_h^{\pi^k}(s) \ge \underline{V}_h^k(s)$. As a result,

$$\begin{aligned}
&|\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2] - \mathbb{V}_h V_{h+1}^{\pi^k}|(s,a,b) \\
=&|[\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)^2/4 - \mathbb{P}_h(V_{h+1}^{\pi^k})^2](s,a,b) \\
&\qquad\qquad - [(\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k))^2/4 - (\mathbb{P}_h V_{h+1}^{\pi^k})^2](s,a,b)| \\
\le&[\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1}^k)^2 - \mathbb{P}_h(\underline{V}_{h+1}^k)^2 - (\widehat{\mathbb{P}}_h^k\underline{V}_{h+1}^k)^2 + (\mathbb{P}_h\overline{V}_{h+1}^k)^2](s,a,b) \\
\le&[|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(\overline{V}_{h+1}^k)^2| + |\mathbb{P}_h[(\overline{V}_{h+1}^k)^2 - (\underline{V}_{h+1}^k)^2]| \\
&\qquad + |(\widehat{\mathbb{P}}_h^k\underline{V}_{h+1}^k)^2 - (\mathbb{P}_h\underline{V}_{h+1}^k)^2| + |(\mathbb{P}_h\underline{V}_{h+1}^k)^2 - (\mathbb{P}_h\overline{V}_{h+1}^k)^2|](s,a,b)
\end{aligned}$$

These terms can be bounded separately by using event E_1:

$$\begin{aligned}
|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(\overline{V}_{h+1}^k)^2|(s,a,b) \le& H^2\|(\widehat{\mathbb{P}}_h^k - \Pr_h)(\cdot \mid s,a,b)\|_1 \\
\le& \mathcal{O}(H^2\sqrt{\frac{S\iota}{\max\{N_h^k(s,a,b),1\}}}), \\
|\mathbb{P}_h[(\overline{V}_{h+1}^k)^2 - (\underline{V}_{h+1}^k)^2]|(s,a,b) \le& 2H[\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)](s,a,b), \\
|(\widehat{\mathbb{P}}_h^k\underline{V}_{h+1}^k)^2 - (\mathbb{P}_h\underline{V}_{h+1}^k)^2|(s,a,b) \le& 2H[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)\underline{V}_{h+1}^k](s,a,b) \\
\le& \mathcal{O}(H^2\sqrt{\frac{S\iota}{\max\{N_h^k(s,a,b),1\}}}), \\
|(\mathbb{P}_h\underline{V}_{h+1}^k)^2 - (\mathbb{P}_h\overline{V}_{h+1}^k)^2|(s,a,b) \le& 2H[\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)](s,a,b).
\end{aligned}$$

Combining with

$$H^2\sqrt{\frac{S\iota}{\max\{N_h^k(s,a,b),1\}}} \le 1 + \frac{H^4 S\iota}{\max\{N_h^k(s,a,b),1\}}$$

completes the proof. □

Finally we come to the proof of Theorem 17.

Proof of Theorem 17. Suppose event E_1 holds. We define Δ_h^k, ζ_h^k abd ξ_h^k as in the proof of Theorem 16. As before we have

$$\begin{aligned}\Delta_h^k \le & \zeta_h^k + \left(1+\frac{c_3}{H}\right)\mathbb{P}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)\left(s_h^k, a_h^k, b_h^k\right) \\ & + 4c_2C\left(\sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2](s_h^k, a_h^k, b_h^k)}{\max\{N_h^k(s_h^k, a_h^k, b_h^k), 1\}}} + \frac{H^2 S\iota}{\max\{N_h^k(s_h^k, a_h^k, b_h^k), 1\}}\right).\end{aligned} \tag{B.12}$$

By Lemma 71,

$$\begin{aligned}& \sqrt{\frac{\iota\widehat{\mathbb{V}}_h^k[(\overline{V}_{h+1}^k + \underline{V}_{h+1}^k)/2](s,a,b)}{\max\{N_h^k(s,a,b),1\}}} \\ \le & \mathcal{O}\left(\sqrt{\frac{\iota\mathbb{V}_h V_{h+1}^{\pi^k}(s,a,b)+\iota}{\max\{N_h^k(s,a,b),1\}}} + \sqrt{\frac{H\iota\,\mathrm{Pr}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b)}{\max\{N_h^k(s,a,b),1\}}} + \frac{H^2\sqrt{S}\iota}{\max\{N_h^k(s,a,b),1\}}\right) \\ \le & c_4\left(\sqrt{\frac{\iota\mathbb{V}_h V_{h+1}^{\pi^k}(s,a,b)+\iota}{\max\{N_h^k(s,a,b),1\}}} + \frac{\mathrm{Pr}_h(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k)(s,a,b)}{H} + \frac{H^2\sqrt{S}\iota}{\max\{N_h^k(s,a,b),1\}}\right),\end{aligned} \tag{B.13}$$

where c_4 is some absolute constant. Define $c_5 := 4c_2c_4C + c_3$ and $\kappa := 1 + c_5/H$. Plugging (B.13) back into (B.12), we have

$$\begin{aligned}& \Delta_h^k \\ \le & \kappa\Delta_{h+1}^k + \kappa\xi_h^k + \zeta_h^k + \mathcal{O}\left(\sqrt{\frac{\iota\mathbb{V}_h V_{h+1}^{\pi^k}(s_h^k, a_h^k, b_h^k)}{N_h^k(s_h^k, a_h^k, b_h^k)}} + \sqrt{\frac{\iota}{N_h^k(s_h^k, a_h^k, b_h^k)}} + \frac{H^2 S\iota}{N_h^k(s_h^k, a_h^k, b_h^k)}\right)\Bigg\}.\end{aligned} \tag{B.14}$$

Recursing this argument for $h \in [H]$ and summing over k,

$$\begin{aligned}& \sum_{k=1}^K \Delta_1^k \\ \le & \sum_{k=1}^K\sum_{h=1}^H\left[\kappa^{h-1}\zeta_h^k + \kappa^h\xi_h^k + \mathcal{O}\left(\sqrt{\frac{\iota\mathbb{V}_h V_{h+1}^{\pi^k}(s_h^k, a_h^k, b_h^k)}{\max\{N_h^k,1\}}} + \sqrt{\frac{\iota}{\max\{N_h^k,1\}}} + \frac{H^2 S\iota}{\max\{N_h^k,1\}}\right)\right].\end{aligned}$$

The remaining steps are the same as that in the proof of Theorem 16 except that we need to bound the sum of variance term.

By Cauchy-Schwarz,

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\sqrt{\frac{\mathbb{V}_h V_{h+1}^{\pi^k}(s_h^k, a_h^k, b_h^k)}{\max\{N_h^k(s_h^k, a_h^k, b_h^k), 1\}}}$$
$$\leq \sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{V}_h V_{h+1}^{\pi^k}(s_h^k, a_h^k, b_h^k) \cdot \sum_{k=1}^{K}\sum_{h=1}^{H}\frac{1}{\max\{N_h^k(s_h^k, a_h^k, b_h^k), 1\}}}.$$

By the Law of total variation and standard martingale concentration (see Lemma C.5 in Jin et al. [2018] for a formal proof), with probability at least $1-p$, we have

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{V}_h V_{h+1}^{\pi^k}(s_h^k, a_h^k, b_h^k) \leq \mathcal{O}\big(HT + H^3\iota\big).$$

Putting all relations together, we obtain that with probability at least $1-2p$ (one p comes from $\Pr(E_1) \geq 1-p$ and the other comes from the inequality for bounding the variance term),

$$\mathfrak{R}(K) = \sum_{k=1}^{K}(V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1) \leq \mathcal{O}(\sqrt{H^2SABT\iota} + H^3S^2AB\iota^2).$$

Rescaling p completes the proof. □

B.2 Proof for Section 3.3 – Reward-Free Learning

B.2.1 Proof of Theorem 18

In this section, we prove Theorem 18 for the single reward function case, i.e., $N = 1$. The proof for multiple reward functions ($N > 1$) simply follows from taking a union bound, that is, replacing the failure probability p by Np.

Let (μ^k, ν^k) be an arbitrary Nash-equilibrium policy of $\widehat{\mathcal{M}}^k := (\widehat{\mathbb{P}}^k, \widehat{r}^k)$, where $\widehat{\mathbb{P}}^k$ and $\widehat{r}^k$ are our empirical estimate of the transition and the reward at the beginning of the k-th episode in Algorithm 2, respectively. We use $N_h^k(s, a, b)$ to denote the number we have visited the state-action tuple (s, a, b) at the h'=th step before the

k-th episode. And the bonus used in the k-th episode can be written as

$$\beta_h^k(s,a,b) := C\left(\sqrt{\frac{H^2\iota}{\max\{N_h^k(s,a,b),1\}}} + \frac{H^2S\iota}{\max\{N_h^k(s,a,b),1\}}\right), \tag{B.15}$$

where $\iota = \log(SABT/p)$ and C is some large absolute constant.

We use $\widehat{Q}^k$ and $\widehat{V}^k$ to denote the empirical optimal value functions of $\widehat{\mathcal{M}}^k$ as following.

$$\begin{cases} \widehat{Q}_h^k(s,a,b) = (\widehat{\mathbb{P}}_h^k\widehat{V}_{h+1})(s,a,b) + \widehat{r}_h^k(s,a,b), \\ \widehat{V}_h^k(s) = \max\limits_{\mu}\min\limits_{\nu}\mathbb{D}_{\mu\times\nu}\widehat{Q}_h^k(s). \end{cases} \tag{B.16}$$

Since (μ^k,ν^k) is a Nash-equilibrium policy of $\widehat{\mathcal{M}}^k$, we also have $\widehat{V}_h^k(s) = \mathbb{D}_{\mu^k\times\nu^k}\widehat{Q}_h^k(s)$.

We begin with stating a useful property of matrix game that will be frequently used in our analysis. Since its proof is quite simple, we omit it here.

Lemma 72. *Let $\mathbf{X},\mathbf{Y},\mathbf{Z} \in \mathbb{R}^{A\times B}$ and Δ_d be the d-dimensional simplex. Suppose $|\mathbf{X}-\mathbf{Y}| \le \mathbf{Z}$, where the inequality is entry-wise. Then*

$$\left|\max_{\mu\in\triangle_A}\min_{\nu\in\triangle_B}\mu^\top\mathbf{X}\nu - \max_{\mu\in\triangle_A}\min_{\nu\in\triangle_B}\mu^\top\mathbf{Y}\nu\right| \le \max_{i,j}\mathbf{Z}_{ij}. \tag{B.17}$$

Lemma 73. *Let c_1 be some large absolute constant such that $c_1^2 + c_1 \le C$. Define event E_1 to be: for all h,s,a,b,s' and $k \in [K]$,*

$$\begin{cases} |[(\widehat{\mathbb{P}}_h^k - \Pr_h)V_{h+1}^\star](s,a,b)| \le \dfrac{c_1}{10}\sqrt{\dfrac{H^2\iota}{\max\{N_h^k(s,a,b),1\}}}, \\ |(\widehat{r}_h^k - r_h)(s,a,b)| \le \dfrac{c_1}{10}\sqrt{\dfrac{H^2\iota}{\max\{N_h^k(s,a,b),1\}}}, \\ |(\widehat{\mathbb{P}}_h^k - \Pr_h)(s' \mid s,a,b)| \le \dfrac{c_1}{10}\left(\sqrt{\dfrac{\widehat{\mathbb{P}}_h^k(s'\mid s,a,b)\iota}{\max\{N_h^k(s,a,b),1\}}} + \dfrac{\iota}{\max\{N_h^k(s,a,b),1\}}\right). \end{cases} \tag{B.18}$$

We have $\Pr(E_1) \ge 1-p$.

Proof. The proof is standard: apply concentration inequalities and then take a union bound. For completeness, we provide the proof of the third one here.

Consider a fixed (s,a,b,h) tuple.

Let's consider the following equivalent random process: (a) before the agent starts, the environment samples $\{s^{(1)}, s^{(2)}, \ldots, s^{(K)}\}$ independently from $\Pr_h(\cdot \mid s,a,b)$; (b) during the interaction between the agent and the environment, the i^{th} time the agent reaches (s,a,b,h), the environment will make the agent transit to $s^{(i)}$. Note that the randomness induced by this interaction procedure is exactly the same as the original one, which means the probability of any event in this context is the same as in the original problem. Therefore, it suffices to prove the target concentration inequality in this 'easy' context. Denote by $\widehat{\mathbb{P}}_h^{(t)}(\cdot \mid s,a,b)$ the empirical estimate of $\Pr_h(\cdot \mid s,a,b)$ calculated using $\{s^{(1)}, s^{(2)}, \ldots, s^{(t)}\}$. For a fixed t and s', by the empirical Bernstein inequality, we have with probability at least $1-p/S^2ABT$,

$$|(\Pr_h - \widehat{\mathbb{P}}_h^{(t)})(s' \mid s,a,b)| \le \mathcal{O}\left(\sqrt{\frac{\widehat{\mathbb{P}}_h^{(t)}(s' \mid s,a,b)\iota}{t}} + \frac{\iota}{t}\right).$$

Now we can take a union bound over all s,a,b,h,s' and $t \in [K]$, and obtain that with probability at least $1-p$, for all s,a,b,h,s' and $t \in [K]$,

$$|(\Pr_h - \widehat{\mathbb{P}}_h^{(t)})(s' \mid s,a,b)| \le \mathcal{O}\left(\sqrt{\frac{\widehat{\mathbb{P}}_h^{(t)}(s' \mid s,a,b)\iota}{t}} + \frac{\iota}{t}\right).$$

Note that the agent can reach each (s,a,b,h) for at most K times, so we conclude the third inequality also holds with probability at least $1-p$. □

The following lemma states that the empirical optimal value functions are close to the true optimal ones, and their difference is controlled by the exploration value functions calculated in Algorithm 2.

Lemma 74. *Suppose event E_1 (defined in Lemma 73) holds. Then for all h,s,a,b*

and $k \in [K]$, we have,

$$\begin{cases} \left|\widehat{Q}_h^k(s,a,b) - Q_h^\star(s,a,b)\right| \le \widetilde{Q}_h^k(s,a,b), \\ \left|\widehat{V}_h^k(s) - V_h^\star(s)\right| \le \widetilde{V}_h^k(s). \end{cases} \tag{B.19}$$

Proof. Let's prove by backward induction on h. The case of $h = H+1$ holds trivially.

Assume the conclusion hold for $(h+1)$-th step. For h-th step,

$$\begin{aligned} &\left|\widehat{Q}_h^k(s,a,b) - Q_h^\star(s,a,b)\right| \\ \le{}& \min\{\left|[(\widehat{\mathbb{P}}_h^k - \Pr_h)V_{h+1}^\star](s,a,b)\right| + |(\widehat{r}_h^k - r_h)(s,a,b)| \\ &\quad + \left|[\widehat{\mathbb{P}}_h^k(\widehat{V}_{h+1}^k - V_{h+1}^\star)](s,a,b)\right|, H\} \\ \overset{(i)}{\le}{}& \min\left\{\beta_h^k(s,a,b) + (\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s,a,b), H\right\} \overset{(ii)}{=} \widetilde{Q}_h^k(s,a,b), \end{aligned} \tag{B.20}$$

where (i) follows from the induction hypothesis and event E_1, and (ii) follows from the definition of $\widetilde{Q}_h^k$. By Lemma 72, we immediately obtain $|\widehat{V}_h^k(s) - V_h^\star(s)| \le \widetilde{V}_h^k(s)$. □

Now, we are ready to establish the key lemma in our analysis using Lemma 74.

Lemma 75. *Suppose event E_1 (defined in Lemma 73) holds. Then for all h, s, a, b and $k \in [K]$, we have*

$$\begin{cases} |\widehat{Q}_h^k(s,a,b) - Q_h^{\dagger,\nu^k}(s,a,b)| \le \alpha_h \widetilde{Q}_h^k(s,a,b), \\ |\widehat{V}_h^k(s) - V_h^{\dagger,\nu^k}(s)| \le \alpha_h \widetilde{V}_h^k(s), \end{cases} \tag{B.21}$$

and

$$\begin{cases} |\widehat{Q}_h^k(s,a,b) - Q_h^{\mu^k,\dagger}(s,a,b)| \le \alpha_h \widetilde{Q}_h^k(s,a,b), \\ |\widehat{V}_h^k(s) - V_h^{\mu^k,\dagger}(s)| \le \alpha_h \widetilde{V}_h^k(s), \end{cases} \tag{B.22}$$

where $\alpha_{H+1} = 0$ and $\alpha_h = [(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}] \le 4$.

Proof. We only prove the first set of inequalities. The second one follows exactly the same. Again, the proof is by performing backward induction on h. It is trivial to see the conclusion holds for $(H+1)$-th step with $\alpha_{H+1} = 0$. Now, assume the conclusion holds for $(h+1)$-th step. For h-th step,

$$\begin{aligned}
&|\widehat{Q}^k_h(s,a,b) - Q^{\dagger,\nu^k}_h(s,a,b)| \\
\le& \min\Big\{ |[(\widehat{\mathbb{P}}^k_h - \Pr_h)(V^{\dagger,\nu^k}_{h+1} - V^\star_{h+1})](s,a,b)| + |(\widehat{\mathbb{P}}^k_h - \Pr_h)V^\star_{h+1}(s,a,b)| \\
&+ |(\widehat{r}^k_h - r_h)(s,a,b)| + |[\widehat{\mathbb{P}}_h(\widehat{V}^k_{h+1} - V^{\dagger,\nu^k}_{h+1})](s,a,b)|, H \Big\} \\
\le& \min\Big\{ \underbrace{|[(\widehat{\mathbb{P}}^k_h - \Pr_h)(V^{\dagger,\nu^k}_{h+1} - V^\star_{h+1})](s,a,b)|}_{(T_1)} + c_1\sqrt{\frac{H^2\iota}{\max\{N^k_h(s,a,b),1\}}} \\
&+ \underbrace{|[\widehat{\mathbb{P}}_h(\widehat{V}^k_{h+1} - V^{\dagger,\nu^k}_{h+1})](s,a,b)|}_{(T_2)}, H \Big\},
\end{aligned} \tag{B.23}$$

where the second inequality follows from the definition of event E_1.

We can control the term (T_1) by combining Lemma 74 and the induction hypothesis to bound $|V^{\dagger,\nu^k}_{h+1} - V^\star_{h+1}|$, and then applying the third inequality in event E_1:

$$\begin{aligned}
(T_1) \le& \sum_{s'} |\widehat{\mathbb{P}}^k_h(s' \mid s,a,b) - \Pr_h(s' \mid s,a,b)||V^{\dagger,\nu^k}_{h+1} - V^\star_{h+1}(s')| \\
\le& \sum_{s'} |\widehat{\mathbb{P}}^k_h(s' \mid s,a,b) - \Pr_h(s' \mid s,a,b)|\Big(|V^{\dagger,\nu^k}_{h+1} - \widehat{V}^k_{h+1}(s')| + |\widehat{V}^k_{h+1} - V^\star_{h+1}(s')|\Big) \\
\le& \sum_{s'} |\widehat{\mathbb{P}}^k_h(s' \mid s,a,b) - \Pr_h(s' \mid s,a,b)|(\alpha_{h+1}+1)\widetilde{V}^k_{h+1} \\
\le& \frac{(\alpha_{h+1}+1)}{H}(\widehat{\mathbb{P}}^k_h\widetilde{V}^k_{h+1})(s,a,b) + \frac{c_1^2(\alpha_{h+1}+1)H^2S\iota}{\max\{N^k_h(s,a,b),1\}}.
\end{aligned} \tag{B.24}$$

The term (T_2) is bounded by directly applying the induction hypothesis

$$|[\widehat{\mathbb{P}}_h(\widehat{V}^k_{h+1} - V^{\dagger,\nu^k}_{h+1})](s,a,b)| \le \alpha_{h+1}[\widehat{\mathbb{P}}_h\widetilde{V}^k_{h+1}](s,a,b). \tag{B.25}$$

Plugging (B.24) and (B.25) into (B.23), we obtain

$$\begin{aligned}
&\left|\widehat{Q}_h^k(s,a,b) - Q_h^{\dagger,\nu^k}(s,a,b)\right| \\
\le& \min\Bigg\{(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}[\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a,b) + c_1\sqrt{\frac{H^2\iota}{\max\{N_h^k(s,a,b),1\}}} \\
&\qquad + \frac{c_1^2(\alpha_{h+1}+1)H^2S\iota}{\max\{N_h^k(s,a,b),1\}}, H\Bigg\} \\
\overset{(i)}{\le}& \min\left\{\left((1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}\right)[\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a,b) + \beta_h^k(s,a,b), H\right\} \\
\overset{(ii)}{\le}& \left((1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}\right)\widetilde{Q}_h^k(s,a,b),
\end{aligned} \tag{B.26}$$

where (i) follows from the definition of β_h^k, and (ii) follows from the definition of $\widetilde{Q}_h^k$. Therefore, by (B.26), choosing $\alpha_h = [(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}]$ suffices for the purpose of induction.

Now, let's prove the inequality for V functions.

$$\begin{aligned}
|(\widehat{V}_h^k - V_h^{\dagger,\nu^k})(s)| &\overset{(i)}{=} |\max_{\mu\in\triangle_A}(\mathbb{D}_{\mu,\nu^k}\widehat{Q}_h^k)(s) - \max_{\mu\in\triangle_A}(\mathbb{D}_{\mu,\nu^k}Q_h^{\dagger,\nu^k})(s)| \\
&\overset{(ii)}{\le} \max_{a,b}\left[\alpha_h\widetilde{Q}_h^k(s,a,b)\right] = \alpha_h\widetilde{V}_h^k(s),
\end{aligned} \tag{B.27}$$

where (i) follows from the definition of $\widehat{V}_h^k$ and $V_h^{\dagger,\nu^k}$, and (ii) uses (B.26) and Lemma 72. □

Theorem 76 (Guarantee for UCB-VI from Azar et al. [2017]). *For any $p \in (0,1]$, choose the exploration bonus β_t in Algrothm 2 as* (B.15). *Then, with probability at least $1-p$,*

$$\sum_{k=1}^K \widetilde{V}_1^k(s_1) \le \mathcal{O}(\sqrt{H^4SAK\iota} + H^3S^2A\iota^2).$$

Proof of Theorem 18. Recall that out $= \arg\min_{k\in[K]} \widetilde{V}_h^k(s)$. By Lemma 75 and Theorem 76, with probability at least $1-2p$,

$$\begin{aligned}
V_h^{\dagger,\nu^{\text{out}}}(s) - V_h^{\mu^{\text{out}},\dagger}(s) \le& |V_h^{\dagger,\nu^{\text{out}}}(s) - \widehat{V}_h^{\text{out}}(s)| + |\widehat{V}_h^{\text{out}}(s) - V_h^{\mu^{\text{out}},\dagger}(s)| \\
\le& 8\widetilde{V}_h^{\text{out}}(s) \le \mathcal{O}(\sqrt{\frac{H^4SA\iota}{K}} + \frac{H^3S^2A\iota^2}{K}).
\end{aligned} \tag{B.28}$$

Rescaling p completes the proof. □

B.2.2 Vanilla Nash Value Iteration

Here, we provide one optional algorithm, Vanilla Nash VI, for computing the Nash equilibrium policy for a *known* model. Its only difference from the value iteration algorithm for MDPs is that the maximum operator is replaced by the minimax operator in Line 7. We remark that the Nash equilibrium for a two-player zero-sum game can be computed in polynomial time.

Algorithm 17 Vanilla Nash Value Iteration

```
1: **Input**: model $\widehat{\mathcal{M}} = (\widehat{\mathbb{P}}, \widehat{r})$.
2: **Initialize:** for all $(s,a,b)$, $V_{H+1}(s,a,b) \leftarrow 0$.
3: **for** step $h = H, H-1, \ldots, 1$ **do**
4:     **for** $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ **do**
5:         $Q_h(s,a,b) \leftarrow [\widehat{\mathbb{P}}_h V_{h+1}](s,a,b) + \widehat{r}_h(s,a,b)$.
6:     **for** $s \in \mathcal{S}$ **do**
7:         $(\widehat{\mu}_h(\cdot \mid s), \widehat{\nu}_h(\cdot \mid s)) \leftarrow \text{NASH-ZERO-SUM}(Q_h(s,\cdot,\cdot))$.
8:         $V_h(s) \leftarrow \widehat{\mu}_h(\cdot \mid s)^\top Q_h(s,\cdot,\cdot)\widehat{\nu}_h(\cdot \mid s)$.
9: **Output** $(\widehat{\mu}, \widehat{\nu}) \leftarrow \{(\widehat{\mu}_h(\cdot \mid s), \widehat{\nu}_h(\cdot \mid s))\}_{(h,s)\in[H]\times\mathcal{S}}$.
```

By recalling the definition of best responses in Section 2.3.2, one can directly see that the output policy $(\widehat{\mu}, \widehat{\nu})$ is a Nash equilibrium for $\widehat{\mathcal{M}}$.

B.2.3 Proof of Theorem 19

In this section, we first prove a $\Theta(AB/\varepsilon^2)$ lower bound for reward-free learning of matrix games, i.e., $S = H = 1$, and then generalize it to $\Theta(SABH^2/\varepsilon^2)$ for reward-free learning of Markov games.

Reward-free learning of matrix games

In the matrix game, let the max-player pick row and the min-player pick column. We consider the following family of Bernoulli matrix games

$$\mathfrak{M}(\varepsilon) := \left\{\mathcal{M}^{a^\star b^\star} \in \mathbb{R}^{A\times B}\right\}, \tag{B.29}$$

where in matrix game $\mathcal{M}^{a^\star b^\star}$, the reward is sampled from Bernoulli$(\mathcal{M}^{a^\star b^\star}_{ab})$ if the max-player picks the a-th row and the min-player picks the b-th column. Here $\mathcal{M}^{a^\star b^\star}_{ab} := \frac{1}{2} + (1 - 2 \cdot \mathbf{1}\{a \neq a^\star \& b = b^\star\})\varepsilon$ for any $(a^\star, b^\star) \in [A] \times [B]$.

$$\begin{array}{cc} & \text{Min-player} \\ \text{Max-player} & \begin{array}{cccccccc} \text{action} & 1 & \dots & b^\star - 1 & b^\star & b^\star + 1 & \dots & B \\ 1 & + & \dots & + & - & + & \dots & + \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a^\star - 1 & + & \dots & + & - & + & \dots & + \\ a^\star & + & \dots & + & + & + & \dots & + \\ a^\star + 1 & + & \dots & + & - & + & \dots & + \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A & + & \dots & + & - & + & \dots & + \end{array} \end{array} \tag{B.30}$$

Above, we visualize $\mathcal{M}^{a^\star b^\star}$ by using $+$ and $-$ to represent $1/2 + \varepsilon$ and $1/2 - \varepsilon$, respectively. It is direct to see that the optimal (Nash equilibrium) policy for the max-player is always picking the $a^\star$-th row. If the max-player picks the $a^\star$-th row with probability smaller than $2/3$, it is at least $\varepsilon/10$ suboptimal.

Lemma 77. *Consider an arbitrary **fixed** matrix game $\mathcal{M}^{a^\star b^\star}$ from $\mathfrak{M}(\varepsilon)$ and $N \in \mathbb{N}$. If there exists an algorithm $\mathcal{A}$ such that when running on $\mathcal{M}^{a^\star b^\star}$, it uses at most N samples and outputs an $\varepsilon/10$-optimal policy with probability at least p, then there exists an algorithm $\widehat{\mathcal{A}}$ that can identify $a^\star$ with probability at least p using at most N samples.*

Proof. We simply define $\widehat{\mathcal{A}}$ as running algorithm $\mathcal{A}$ and choosing the most played row by its output policy as the guess for $a^\star$. Because any $\varepsilon/10$-optimal policy must play $a^\star$ with probability at least $2/3$, we obtain $\widehat{\mathcal{A}}$ will correctly identify $a^\star$ with probability at least p. □

Lemma 77 directly implies that in order to prove the desired lower bound for reward-free matrix games:

Claim 1. *for **any** reward-free algorithm $\mathcal{A}$ using at most $N = AB/(10^3\varepsilon^2)$ samples, there exists a matrix game $\mathcal{M}^{a^\star b^\star}$ in $\mathfrak{M}(\varepsilon)$ such that when running $\mathcal{A}$ on $\mathcal{M}^{a^\star b^\star}$, it will output a policy that is **at least** $\varepsilon/10$ suboptimal for the max-player with probability at least* $1/4$*,*

it suffices to prove the following claim:

Claim 2. *for **any** reward-free algorithm $\widehat{\mathcal{A}}$ using at most $N = AB/(10^3\varepsilon^2)$ samples, there exists a matrix game $\mathcal{M}^{a^\star b^\star}$ in $\mathfrak{M}(\varepsilon)$ such that when running $\widehat{\mathcal{A}}$ on $\mathcal{M}$, it will fail to identify the optimal row with probability at least* $1/4$*.*

Remark 78. *By Lemma 77, the existence of such 'ideal' $\mathcal{A}$ implies the existence of an 'ideal' $\widehat{\mathcal{A}}$, so to prove such 'ideal' $\mathcal{A}$ does not exist (Claim 1), it suffices to show such 'ideal' $\widehat{\mathcal{A}}$ does not exist (Claim 2).*

Proof of Claim 2. WLOG, we assume $\widehat{\mathcal{A}}$ is deterministic. Since $\widehat{\mathcal{A}}$ is *reward-free*, being deterministic means that in the exploration phase algorithm $\widehat{\mathcal{A}}$ always pulls each arm (a, b) for some *fixed* $n(a, b)$ times (because there is no information revealed in this phase), and in the planning phase it outputs a guess for $a^\star$, which is a deterministic function of the reward information revealed.

We define the following notations:

- L: the stochastic reward information revealed after algorithm $\widehat{\mathcal{A}}$'s pulling.
- $\Pr_\star$: the probability measure induced by picking $\mathcal{M}^{a^\star b^\star}$ uniformly at random from $\mathfrak{M}(\varepsilon)$ and then running $\widehat{\mathcal{A}}$ on $\mathcal{M}$.
- $\Pr_{ab}$: the probability measure induced by running $\widehat{\mathcal{A}}$ on $\mathcal{M}^{ab}$.
- $\Pr_{0b}$: the probability measure induced by running $\mathcal{A}$ on matrix $\mathcal{M}^{0b}$, whose b-th column are all $(1/2 - \varepsilon)$'s and other columns are all $(1/2 + \varepsilon)$'s.[2]
- $f(L)$: the output of $\widehat{\mathcal{A}}$ based on the stochastic reward information L revealed. More precisely, f is function mapping from $[0, 1]^N$ to $[A]$.

[2] We comment that matrix $\mathcal{M}^{0b}$ does not belong to $\mathfrak{M}(\varepsilon)$.

We have

$$\begin{aligned}\Pr_{\star}(f(L) \neq a^\star) &\geq \frac{1}{AB}\sum_{a,b}\Pr_{0b}(f(L)\neq a) - \frac{1}{AB}\sum_{a,b}\|\Pr_{ab}(L=\cdot) - \Pr_{0b}(L=\cdot)\|_1 \\ &\geq 1 - \frac{1}{A} - \frac{1}{AB}\sum_{a,b}\sqrt{2\mathrm{KL}(\Pr_{0b}\|\Pr_{ab})} \\ &= 1 - \frac{1}{A} - \frac{1}{AB}\sum_{a,b}\sqrt{2n(a,b)[(\frac{1}{2}-\varepsilon)\log\frac{\frac{1}{2}-\varepsilon}{\frac{1}{2}+\varepsilon} + (\frac{1}{2}+\varepsilon)\log\frac{\frac{1}{2}+\varepsilon}{\frac{1}{2}-\varepsilon}]} \\ &\geq 1 - \frac{1}{A} - \frac{10}{AB}\sum_{a,b}\sqrt{n(a,b)\varepsilon^2} \\ &\geq 1 - \frac{1}{A} - \sqrt{\frac{100N\varepsilon^2}{AB}},\end{aligned} \tag{B.31}$$

where the second inequality follows from $\sum_{a,b}\Pr_{0b}(f(L)\neq a) = \sum_{a,b}[1 - \Pr_{0b}(f(L) = a)] = B(A-1)$ and Pinsker's inequality. Finally, plugging in $N = AB/(10^3\varepsilon^2)$ concludes the proof. □

Remark 79. *The arguments in proving Claim 2 basically follows the same line in proving lower bounds for multi-arm bandits [e.g., see Lattimore and Szepesvári, 2020].*

Reward-free learning of Markov games

Now let's generalize the $\Theta(AB/\varepsilon^2)$ lower bound for reward-free learning of matrix games to $\Theta(SABH^2/\varepsilon^2)$ for reward-free learning of Markov games. We can follow the same way of generalizing a lower bound for multi-arm bandits to a lower bound for MDPs [see e.g., Dann and Brunskill, 2015, Lattimore and Szepesvári, 2020, Zhang et al., 2020b].

Proof sketch. Given the family of Bernoulli matrix games $\mathfrak{M}(\cdot)$ defined in (B.29), we simply construct a Markov game to consist of SH Bernoulli matrix games $\{M^{s,h}\}_{(s,h)\in[S]\times[H]}$ where $M^{s,h}$'s are sampled independently and identically from the uniform distribution over $\mathfrak{M}(\varepsilon/H)$. We will define the transition measure to be totally 'uniform at random' so that in each episode the agent will always reach each $M^{s,h}$ with probability $1/S$ (it is not $1/(SH)$ because in each episode the agent can visit H matrix games). As a result, to guarantee ε-optimality, the output policy must be at least $2\varepsilon/H$-optimal for at least $SH/2$ different $M^{s,h}$'s. Recall Claim 1 shows learning a $2\varepsilon/H$-optimal policy for a single $M^{s,h}$ requires $\Omega(H^2AB/\varepsilon^2)$ samples. Therefore, we need $\Omega(H^3AB/\varepsilon^2)$ samples in total for learning $SH/2$ different $M^{s,h}$'s.

Below, we provide a rigorous proof where the constants may be slightly different from those in our sketch. We remark that although the notations we will use are involved, they are only introduced for rigorousness and there is no real technical difficulty or new informative idea in the following proof.

Construction We define the following family of Markov games:

$$\mathfrak{J}(\varepsilon) := \left\{\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star) : \ (\boldsymbol{a}^\star, \boldsymbol{b}^\star) \in [A]^{H\times S} \times [B]^{H\times S}\right\}, \tag{B.32}$$

where MG $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ is defined as

- **States and actions:** $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ is a finite-horizon MG with $S+1$ states and of length $H+1$. There is a fixed initial state s_0 in the first step, S states $\{s_1, \ldots, s_S\}$ in the remaining steps. The two players have A and B actions, respectively.

- **Rewards:** there is no reward in the first step. For the remaining steps $h \in \{2, \ldots, H+1\}$, if the agent takes action (a, b) at state s_i in the hth step, it will receive a binary reward sampled from

$$\mathrm{Bernoulli}\Big(\frac{1}{2} + (1 - 2\cdot \mathbf{1}\{a \neq \boldsymbol{a}^\star_{h-1,i} \& b = \boldsymbol{b}^\star_{h-1,i}\})\frac{\varepsilon}{H}\Big)$$

- **Transitions:** The agent always starts at a fixed initial state s_0 in the first step Regardless of the current state, actions and index of steps, the agent will always transit to one of $s_1, \ldots, s_S$ uniformly at random.

It is direct to see that $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ is a collection of SH matrix games from $\mathfrak{M}(\varepsilon/H)$. Therefore, the optimal policy for the max-player is to always pick action $\boldsymbol{a}^\star_{h-1,i}$ whenever it reaches state s_i at step h ($h \geq 2$).

Formal proof of Theorem 19. Now, let's use $\mathfrak{J}(\varepsilon)$ to prove the $\Theta(SABH^2/\varepsilon^2)$ lower bound (in terms of number of episodes) for reward-free learning of Markov games. We start by proving an analogue of Lemma 77.

Lemma 80. *Consider an arbitrary fixed matrix game $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ from $\mathfrak{J}(\varepsilon)$ and $N \in \mathbb{N}$. If an algorithm $\mathcal{A}$ can output a policy that is at most $\varepsilon/10^3$ suboptimal with probability at least p using at most N samples, then there exists an algorithm $\widehat{\mathcal{A}}$ that can correctly identify at least $SH - \lfloor SH/500 \rfloor$ entries of $\boldsymbol{a}^\star$ with probability at least p using at most N samples.*

Proof. Denote by π the output policy for the max player. Denote by Z the collection of (h,i)'s in $[H] \times [S]$ such that $\pi_{h+1}(\boldsymbol{a}^\star_{h,i} \mid s_i) \le 2/3$.

Observe that each time the max player picks a suboptimal action, it will incur an $2\varepsilon/H$ suboptimality in expectation. As a result, if π is at most $\varepsilon/10^3$-suboptimal, we must have

$$\frac{1}{S} \sum_{(h,i)\in Z} (1 - \pi_{h+1}(\boldsymbol{a}^\star_{h,i} \mid s_i)) \times \frac{2\varepsilon}{H} \le \frac{\varepsilon}{10^3},$$

which implies $|Z| \le SH/500$, that is, for at most $\lfloor SH/500 \rfloor$ different (h,i)'s, $\pi_{h+1}(\boldsymbol{a}^\star_{h,i} \mid s_i) \le 2/3$. Therefore, we can simply pick $\arg\max_a \pi_{h+1}(a \mid s_i)$ as the guess for $\boldsymbol{a}^\star_{h,i}$. Since policy π is at most $\varepsilon/10^3$ suboptimal with probability at least p, our guess will be correct for at least $SH - \lfloor SH/500 \rfloor$ different (s,h) pairs also with probability no smaller than p. □

Similar to the funtion of Lemma 77, Lemma 80 directly implies that in order to prove the desired lower bound for reward-free learning of Markov games:

Claim 3. *for any reward-free algorithm $\mathcal{A}$ that interacts with the environment for at most $K = SABH^2/(10^4\varepsilon^2)$ episodes, there exists $\mathcal{J} \in \mathfrak{J}(\varepsilon)$ such that when running $\mathcal{A}$ on $\mathcal{J}$, it will output a policy that is at least $\varepsilon/10^3$ suboptimal for the max-player with probability at least $1/4$,*

it suffices to prove the following claim:

Claim 4. *for any reward-free learning algorithm $\widehat{\mathcal{A}}$ that interacts with the environment for at most $K = ABSH^2/(10^4\varepsilon^2)$ episodes, there exists $\mathcal{J} \in \mathfrak{J}(\varepsilon)$ such that when running $\widehat{\mathcal{A}}$ on $\mathcal{J}$, it will fail to correctly identify $\boldsymbol{a}^\star_{h,i}$ for at least $\lfloor SH/500 \rfloor + 1$ different (h, i) pairs with probability at least $1/4$.*

Proof of Claim 4. Denote by $\Pr_\star$ ($\mathbb{E}_\star$) the probability measure (expectation) induced by picking $\mathcal{J}$ uniformly at random from $\mathfrak{J}(\varepsilon)$ and then running $\widehat{\mathcal{A}}$ on $\mathcal{J}$. Denote by n_{wrong} the number of (s, h) pairs for which $\widehat{\mathcal{A}}$ fails to identify the optimal actions. Denote by $\text{error}_{h,i}$ the indicator function of the event that $\widehat{\mathcal{A}}$ fails to identify the optimal action for $(h + 1, i)$.

We prove by contradiction. Suppose for any $\mathcal{J} \in \mathfrak{J}(\varepsilon)$, $\widehat{\mathcal{A}}$ can identify the optimal actions for at least $SH - \lfloor SH/500 \rfloor$ different (s, h) pairs with probability larger than $3/4$. Then we have

$$\mathbb{E}_\star[n_{\text{wrong}}] \le \frac{1}{4} \times SH + \frac{3}{4} \times \left\lfloor \frac{SH}{500} \right\rfloor \le \frac{101SH}{400}.$$

Since $\sum_{(h,i)\in[H]\times[S]} \mathbb{E}_\star[\text{error}_{h,i}] = \mathbb{E}_\star[n_{\text{wrong}}]$, there must exists $(h', i') \in [H] \times [S]$ such that $\mathbb{E}_\star[\text{error}_{h',i'}] \le 101/400$. However, in the following, we show that for every $(h, i) \in [H] \times [S]$, $\mathbb{E}_\star[\text{error}_{h,i}] \ge 1/3$. As a result, we obtain a contraction and Claim 4 holds. In the remainder of this section, we will prove for every $(h, i) \in [H] \times [S]$, $\mathbb{E}_\star[\text{error}_{h,i}] \ge 1/3$.

WLOG, we assume $\widehat{\mathcal{A}}$ is deterministic. It suffices to consider an arbitrary *fixed* (h', i') pair and prove $\mathbb{E}_\star[\text{error}_{h',i'}] \ge 1/3$.

For technical reason, we introduce a new MG $\mathcal{J}_{-(h',i')}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ as below:

- **States, actions and transitions:** same as $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$.

- **Rewards:** there is no reward in the first step. For the remaining steps $h \in \{2, \ldots, H + 1\}$, if the agent takes action (a, b) at state s_i in the h^{th} step such that $(h - 1, i) \ne (h', i')$, it will receive a binary reward sampled from

$$\text{Bernoulli}\Big(\frac{1}{2} + (1 - 2 \cdot \mathbf{1}\{a \ne \boldsymbol{a}^\star_{h-1,i} \& b = \boldsymbol{b}^\star_{h-1,i}\})\frac{\varepsilon}{H}\Big),$$

otherwise it will receive a binary reward sampled from

$$\text{Bernoulli}\Big(\frac{1}{2} + (1 - 2 \cdot \mathbf{1}\{b = \boldsymbol{b}^\star_{h-1,i}\})\frac{\varepsilon}{H}\Big).$$

Remark 81. *Briefly speaking, $\mathcal{J}_{-(h',i')}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ is the same as $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ except the matrix game embedded at state $s_{i'}$ at step $h'+1$, where for the max player all its actions are equivalently bad* [3]*. Finally, we remark that $\mathcal{J}_{-(h',i')}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ is* ***independent*** *of $\boldsymbol{a}^\star_{h',i'}$.*

To proceed, we introduce (and recall) the following notations:

- $n(a, b)$: the number of times $\widehat{\mathcal{A}}$ picks action (a, b) at state $s_{i'}$ at step $(h' + 1)$ within K episode.

- $\Pr_{\boldsymbol{a}^\star \boldsymbol{b}^\star}$ ($\mathbb{E}_{\boldsymbol{a}^\star \boldsymbol{b}^\star}$): the probability measure (expectation) induced by running algorithm $\widehat{\mathcal{A}}$ on $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$.

- $\Pr^-_{\boldsymbol{a}^\star \boldsymbol{b}^\star}$ ($\mathbb{E}^-_{\boldsymbol{a}^\star \boldsymbol{b}^\star}$): the probability measure (expectation) induced by running algorithm $\widehat{\mathcal{A}}$ on $\mathcal{J}_{-(h',i')}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$.

- $\Pr_\star$ ($\mathbb{E}_\star$) the probability measure (expectation) induced by picking $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ uniformly at random from $\mathfrak{J}(\varepsilon)$ and running $\widehat{\mathcal{A}}$ on $\mathcal{J}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$.

- L: the whole interaction trajectory of states, actions and rewards produced by algorithm $\widehat{\mathcal{A}}$ within K episodes.

- $f(L)$: the guess of $\widehat{\mathcal{A}}$ for $\boldsymbol{a}^\star_{h',i'}$ based on L.

The key observation here is that for any $(a, b) \in [A] \times [B]$ and $(\boldsymbol{a}^\star, \boldsymbol{b}^\star) \in [A]^{H\times S} \times [B]^{H\times S}$, the expectation $\mathbb{E}^-_{\boldsymbol{a}^\star \boldsymbol{b}^\star}[n(a, b)]$ is independent of $(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$ because the agent does not receive any reward information when interacting with the environment and the transition dynamics of different $\mathcal{J}_{-(h',i')}(\boldsymbol{a}^\star, \boldsymbol{b}^\star)$'s are exactly the same. For simplicity of notation, we denote this expectation by $m(a, b)$. Moreover, note that

[3] A graphic illustration based on (B.30) would be replacing the column $[-, \ldots, -, +, -, \ldots, -]^\top$ with a column of all $-$'s in the matrix game embedded at state $s_{i'}$ at step $h' + 1$.

$\sum_{a,b} m(a,b) = K/S$ because the agent always reach state $s_{i'}$ in step $(h'+1)$ with probability $1/S$ regardless of the actions taken.

By mimicking the arguments in (B.31), we have

$$
\begin{aligned}
&\mathbb{E}_\star[\text{error}_{h',i'}] = \Pr_\star(f(L) \neq \boldsymbol{a}^\star_{h',i'}) \\
&= \frac{1}{(AB)^{SH}} \sum_{(\boldsymbol{a}^\star,\boldsymbol{b}^\star)\in[A]^{H\times S}\times[B]^{H\times S}} \Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}(f(L) \neq \boldsymbol{a}^\star_{h',i'}) \\
&\geq \frac{1}{(AB)^{SH}} \sum_{\boldsymbol{a}^\star,\boldsymbol{b}^\star} \left(\bar{\Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}}(f(L) \neq \boldsymbol{a}^\star_{h',i'}) - \left\| \bar{\Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}}(L=\cdot) - \Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}(L=\cdot) \right\|_1 \right) \\
&= 1 - \frac{1}{A} - \frac{1}{(AB)^{SH}} \sum_{\boldsymbol{a}^\star,\boldsymbol{b}^\star} \left\| \bar{\Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}}(L=\cdot) - \Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}(L=\cdot) \right\| \\
&\geq 1 - \frac{1}{A} - \frac{1}{(AB)^{SH}} \sum_{\boldsymbol{a}^\star,\boldsymbol{b}^\star} \sqrt{2\mathrm{KL}(\bar{\Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}}(L=\cdot), \Pr_{\boldsymbol{a}^\star\boldsymbol{b}^\star}(L=\cdot))} \\
&= 1 - \frac{1}{A} - \frac{1}{(AB)^{SH}} \sum_{\boldsymbol{a}^\star,\boldsymbol{b}^\star} \sqrt{2m(\boldsymbol{a}^\star_{h',i'}, \boldsymbol{b}^\star_{h',i'})\big[(\frac{1}{2}-\frac{\varepsilon}{H})\log\frac{\frac{1}{2}-\frac{\varepsilon}{H}}{\frac{1}{2}+\frac{\varepsilon}{H}} + (\frac{1}{2}+\frac{\varepsilon}{H})\log\frac{\frac{1}{2}+\frac{\varepsilon}{H}}{\frac{1}{2}-\frac{\varepsilon}{H}}\big]} \\
&= 1 - \frac{1}{A} - \frac{1}{AB} \sum_{(a,b)\in[A]\times[B]} \sqrt{2m(a,b)\big[(\frac{1}{2}-\frac{\varepsilon}{H})\log\frac{\frac{1}{2}-\frac{\varepsilon}{H}}{\frac{1}{2}+\frac{\varepsilon}{H}} + (\frac{1}{2}+\frac{\varepsilon}{H})\log\frac{\frac{1}{2}+\frac{\varepsilon}{H}}{\frac{1}{2}-\frac{\varepsilon}{H}}\big]} \\
&\geq 1 - \frac{1}{A} - \frac{10}{AB} \sum_{a,b} \sqrt{m(a,b)\frac{\varepsilon^2}{H^2}} \\
&\geq 1 - \frac{1}{A} - \frac{10\varepsilon}{ABH} \sqrt{AB\sum_{a,b} m(a,b)} = 1 - \frac{1}{A} - \sqrt{\frac{100K\varepsilon^2}{SABH^2}}.
\end{aligned}
\tag{B.33}
$$

Plugging in $K = SABH^2/(10^4\varepsilon^2)$ completes the proof. □

B.3 Proof for Section 3.4.2 – Multi-player General-sum Markov Games

B.3.1 Proof of Theorem 20

NE Version In this part, we prove Theorem 20 (NE version). As before, we begin with proving the optimistic estimations are indeed upper bounds of the corresponding

V-value and Q-value functions.

Lemma 82. *With probability $1-p$, for any $(s, \boldsymbol{a}, h, i)$ and $k \in [K]$:*

$$\overline{Q}_{h,i}^{k}(s, \boldsymbol{a}) \geq Q_{h,i}^{\dagger, \pi_{-i}^{k}}(s, \boldsymbol{a}), \quad \underline{Q}_{h,i}^{k}(s, \boldsymbol{a}) \leq Q_{h,i}^{\pi^{k}}(s, \boldsymbol{a}), \tag{B.34}$$

$$\overline{V}_{h,i}^{k}(s) \geq V_{h,i}^{\dagger, \pi_{-i}^{k}}(s), \quad \underline{V}_{h,i}^{k}(s) \leq V_{h,i}^{\pi^{k}}(s). \tag{B.35}$$

Proof. For each fixed k, we prove this by induction from $h = H+1$ to $h = 1$. For the base case, we know at the $(H+1)$-th step,$\overline{V}_{H+1,i}^{k}(s) = V_{H+1,i}^{\dagger, \pi_{-i}^{k}}(s) = 0$. Now, assume the inequality (B.35) holds for the $(h+1)$-th step, for the h-th step, by the definition of Q-functions,

$$\begin{aligned} \overline{Q}_{h,i}^{k}(s, \boldsymbol{a}) - Q_{h,i}^{\dagger, \pi_{-i}^{k}}(s, \boldsymbol{a}) &= \left[\widehat{\mathbb{P}}_{h}^{k} \overline{V}_{h+1,i}^{k}\right](s, \boldsymbol{a}) - \left[\mathbb{P}_{h} V_{h+1,i}^{\dagger, \pi_{-i}^{k}}\right](s, \boldsymbol{a}) + \beta_t \\ &= \underbrace{\widehat{\mathbb{P}}_{h}^{k}\left(\overline{V}_{h+1,i}^{k} - V_{h+1,i}^{\dagger, \pi_{-i}^{k}}\right)(s, \boldsymbol{a})}_{(A)} + \underbrace{\left(\widehat{\mathbb{P}}_{h}^{k} - \mathbb{P}_{h}\right) V_{h+1,i}^{\dagger, \pi_{-i}^{k}}(s, \boldsymbol{a})}_{(B)} + \beta_t. \end{aligned}$$

By induction hypothesis, for any s', $\left(\overline{V}_{h+1,i}^{k} - V_{h+1,i}^{\dagger, \pi_{-i}^{k}}\right)(s') \geq 0$, and thus $(A) \geq 0$. By uniform concentration [e.g., Lemma 12 in Bai and Jin, 2020], $(B) \leq C\sqrt{SH^2\iota / N_h^k(s, \boldsymbol{a})} = \beta_t$. Putting everything together we have $\overline{Q}_{h,i}^{k}(s, \boldsymbol{a}) - Q_{h,i}^{\dagger, \pi_{-i}^{k}}(s, \boldsymbol{a}) \geq 0$. The second inequality can be proved similarly.

Now assume inequality (B.34) holds for the h-th step, by the definition of V-functions and Nash equilibrium,

$$\overline{V}_{h,i}^{k}(s) = \mathbb{D}_{\pi^k} \overline{Q}_{h,i}^{k}(s) = \max_{\mu} \mathbb{D}_{\mu \times \pi_{-i}^{k}} \overline{Q}_{h,i}^{k}(s).$$

By Bellman equation,

$$V_{h,i}^{\dagger, \pi_{-i}^{k}}(s) = \max_{\mu} \mathbb{D}_{\mu \times \pi_{-i}^{k}} Q_{h,i}^{\dagger, \pi_{-i}^{k}}(s).$$

Since by induction hypothesis, for any $(s, \boldsymbol{a})$, $\overline{Q}_{h,i}^{k}(s, \boldsymbol{a}) \geq Q_{h,i}^{\dagger, \pi_{-i}^{k}}(s, \boldsymbol{a})$. As a result, we also have $\overline{V}^k_{h,i}(s) \geq V^{\dagger,\pi^k_{-i}}_{h,i}(s)$, which is exactly inequality (B.35) for the h-th step. The second inequality can be proved similarly. □

Proof of Theorem 20. Let us focus on the i-th player and ignore the subscript when there is no confusion. To bound

$$\max_i \left(V^{\dagger,\pi^k_{-i}}_{1,i} - V^{\pi^k}_{1,i}\right)\left(s^k_h\right) \leq \max_i \left(\overline{V}^k_{1,i} - \underline{V}^k_{1,i}\right)\left(s^k_h\right),$$

we notice the following propogation:

$$\begin{cases} (\overline{Q}^k_{h,i} - \underline{Q}^k_{h,i})(s,\boldsymbol{a}) \leq \widehat{\mathbb{P}}^k_h(\overline{V}^k_{h+1,i} - \underline{V}^k_{h+1,i})(s,\boldsymbol{a}) + 2\beta^k_h(s,\boldsymbol{a}), \\ (\overline{V}_{h,i} - \underline{V}_{h,i})(s) = [\mathbb{D}_{\pi_h}(\overline{Q}^k_{h,i} - \underline{Q}^k_{h,i})](s). \end{cases} \tag{B.36}$$

We can define $\widetilde{Q}^k_h$ and $\widetilde{V}^k_h$ recursively by $\widetilde{V}^k_{H+1} = 0$ and

$$\begin{cases} \widetilde{Q}^k_h(s,\boldsymbol{a}) = \widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1}(s,\boldsymbol{a}) + 2\beta^k_h(s,\boldsymbol{a}), \\ \widetilde{V}^k_h(s) = [\mathbb{D}_{\pi_h}\widetilde{Q}^k_h](s). \end{cases} \tag{B.37}$$

Then we can prove inductively that for any k, h, s and $\boldsymbol{a}$ we have

$$\begin{cases} \max_i (\overline{Q}^k_{h,i} - \underline{Q}^k_{h,i})(s,\boldsymbol{a}) \leq \widetilde{Q}^k_h(s,\boldsymbol{a}), \\ \max_i (\overline{V}_{h,i} - \underline{V}_{h,i})(s) \leq \widetilde{V}^k_h(s). \end{cases} \tag{B.38}$$

Thus we only need to bound $\sum_{k=1}^K \widetilde{V}^k_1(s)$. Define the shorthand notation

$$\begin{cases} \beta^k_h := \beta^k_h(s^k_h, \boldsymbol{a}^k_h), \\ \Delta^k_h := \widetilde{V}^k_h(s^k_h), \\ \zeta^k_h := [\mathbb{D}_{\pi^k}\widetilde{Q}^k_h]\left(s^k_h\right) - \widetilde{Q}^k_h(s^k_h, \boldsymbol{a}^k_h), \\ \xi^k_h := [\mathbb{P}_h \widetilde{V}^k_{h+1}](s^k_h, \boldsymbol{a}^k_h) - \Delta^k_{h+1}. \end{cases} \tag{B.39}$$

We can check ζ_h^k and ξ_h^k are martingale difference sequences. As a result,

$$\begin{aligned}\Delta_h^k =&\mathbb{D}_{\pi^k}\widetilde{Q}_h^k\left(s_h^k\right)\\ =&\zeta_h^k+\widetilde{Q}_h^k\left(s_h^k,\boldsymbol{a}_h^k\right)\\ =&\zeta_h^k+2\beta_h^k+[\widehat{\mathbb{P}}_h^k\widetilde{V}_{h+1}^k]\left(s_h^k,\boldsymbol{a}_h^k\right)\\ \leq&\zeta_h^k+3\beta_h^k+[\mathbb{P}_h\widetilde{V}_{h+1}^k]\left(s_h^k,\boldsymbol{a}_h^k\right)\\ =&\zeta_h^k+3\beta_h^k+\xi_h^k+\Delta_{h+1}^k.\end{aligned}$$

Recursing this argument for $h \in [H]$ and taking the sum,

$$\sum_{k=1}^K \Delta_1^k \leq \sum_{k=1}^K \left(\zeta_h^k+3\beta_h^k+\xi_h^k\right) \leq O\left(S\sqrt{H^3T\iota\prod_{i=1}^M A_i}\right).$$

□

CCE Version The proof is very similar to the NE version. Specifically, the only part that uses the properties of NE there is Lemma 82. We prove a counterpart here.

Lemma 83. *With probability* $1-p$, *for any* $(s,\boldsymbol{a},h,i)$ *and* $k\in[K]$*:*

$$\overline{Q}_{h,i}^k\left(s,\boldsymbol{a}\right) \geq Q_{h,i}^{\dagger,\pi_{-i}^k}\left(s,\boldsymbol{a}\right),\quad \underline{Q}_{h,i}^k\left(s,\boldsymbol{a}\right) \leq Q_{h,i}^{\pi^k}\left(s,\boldsymbol{a}\right), \tag{B.40}$$

$$\overline{V}_{h,i}^k\left(s\right) \geq V_{h,i}^{\dagger,\pi_{-i}^k}\left(s\right),\quad \underline{V}_{h,i}^k\left(s\right) \leq V_{h,i}^{\pi^k}\left(s\right). \tag{B.41}$$

Proof. For each fixed k, we prove this by induction from $h=H+1$ to $h=1$. For the base case, we know at the $(H+1)$-th step, $\overline{V}_{H+1,i}^k\left(s\right)=V_{H+1,i}^{\dagger,\pi_{-i}^k}\left(s\right)=0$. Now, assume the inequality (B.35) holds for the $(h+1)$-th step, for the h-th step, by the definition of Q-functions,

$$\begin{aligned}\overline{Q}_{h,i}^k\left(s,\boldsymbol{a}\right)-Q_{h,i}^{\dagger,\pi_{-i}^k}\left(s,\boldsymbol{a}\right)=&\left[\widehat{\mathbb{P}}_h^k\overline{V}_{h+1,i}^k\right]\left(s,\boldsymbol{a}\right)-\left[\mathbb{P}_hV_{h+1,i}^{\dagger,\pi_{-i}^k}\right]\left(s,\boldsymbol{a}\right)+\beta_t\\ =&\underbrace{\widehat{\mathbb{P}}_h^k\left(\overline{V}_{h+1,i}^k-V_{h+1,i}^{\dagger,\pi_{-i}^k}\right)\left(s,\boldsymbol{a}\right)}_{(A)}+\underbrace{\left(\widehat{\mathbb{P}}_h^k-\mathbb{P}_h\right)V_{h+1,i}^{\dagger,\pi_{-i}^k}\left(s,\boldsymbol{a}\right)}_{(B)}+\beta_t.\end{aligned}$$

By induction hypothesis, for any s', $\left(\overline{V}^k_{h+1,i} - V^{\dagger,\pi^k_{-i}}_{h+1,i}\right)(s') \geq 0$, and thus $(A) \geq 0$. By uniform concentration, $(B) \leq C\sqrt{SH^2\iota/N^k_h(s,\boldsymbol{a})} = \beta_t$. Putting everything together we have $\overline{Q}^k_{h,i}(s,\boldsymbol{a}) - Q^{\dagger,\pi^k_{-i}}_{h,i}(s,\boldsymbol{a}) \geq 0$. The second inequality can be proved similarly.

Now assume inequality (B.40) holds for the h-th step, by the definition of V-functions and CCE,

$$\overline{V}^k_{h,i}(s) = \mathbb{D}_{\pi^k}\overline{Q}^k_{h,i}(s) \geq \max_{\mu}\mathbb{D}_{\mu\times\pi^k_{-i}}\overline{Q}^k_{h,i}(s).$$

By Bellman equation,

$$V^{\dagger,\pi^k_{-i}}_{h,i}(s) = \max_{\mu}\mathbb{D}_{\mu\times\pi^k_{-i}}Q^{\dagger,\pi^k_{-i}}_{h,i}(s).$$

Since by induction hypothesis, for any $(s,\boldsymbol{a})$, $\overline{Q}^k_{h,i}(s,\boldsymbol{a}) \geq Q^{\dagger,\pi^k_{-i}}_{h,i}(s,\boldsymbol{a})$. As a result, we also have $\overline{V}^k_{h,i}(s) \geq V^{\dagger,\pi^k_{-i}}_{h,i}(s)$, which is exactly inequality (B.35) for the h-th step. The second inequality can be proved similarly. □

CE Version The proof is very similar to the NE version. Specifically, the only part that uses the properties of NE there is Lemma 82. We prove a counterpart here.

Lemma 84. *With probability $1-p$, for any $(s,\boldsymbol{a},h,i)$ and $k \in [K]$:*

$$\overline{Q}^k_{h,i}(s,\boldsymbol{a}) \geq \max_{\phi\in\Phi_i}Q^{\phi\diamond\pi^k}_{h,i}(s,\boldsymbol{a}), \quad \underline{Q}^k_{h,i}(s,\boldsymbol{a}) \leq Q^{\pi^k}_{h,i}(s,\boldsymbol{a}), \tag{B.42}$$

$$\overline{V}^k_{h,i}(s) \geq \max_{\phi\in\Phi_i}V^{\phi\diamond\pi^k}_{h,i}(s), \quad \underline{V}^k_{h,i}(s) \leq V^{\pi^k}_{h,i}(s). \tag{B.43}$$

Proof. For each fixed k, we prove this by induction from $h = H+1$ to $h = 1$. For the base case, we know at the $(H+1)$-th step, $\overline{V}^k_{H+1,i}(s) = \max_{\phi}V^{\phi\diamond\pi^k}_{H+1,i}(s) = 0$. Now, assume the inequality (B.35) holds for the $(h+1)$-th step, for the h-th step, by definition of Q-functions,

$$\begin{aligned}
&\overline{Q}_{h,i}^{k}(s,\boldsymbol{a})-\max_{\phi}Q_{h,i}^{\phi\diamond\pi^k}(s,\boldsymbol{a})\\
=&\left[\widehat{\mathbb{P}}_h^k\overline{V}_{h+1,i}^{k}\right](s,\boldsymbol{a})-\left[\mathbb{P}_h\max_{\phi}V_{h+1,i}^{\phi\diamond\pi^k}\right](s,\boldsymbol{a})+\beta_t\\
=&\underbrace{\widehat{\mathbb{P}}_h^k\left(\overline{V}_{h+1,i}^{k}-\max_{\phi}V_{h+1,i}^{\phi\diamond\pi^k}\right)(s,\boldsymbol{a})}_{(A)}+\underbrace{\left(\widehat{\mathbb{P}}_h^k-\mathbb{P}_h\right)\max_{\phi}V_{h+1,i}^{\phi\diamond\pi^k}(s,\boldsymbol{a})}_{(B)}+\beta_t.
\end{aligned}$$

By induction hypothesis, for any s', $\left(\overline{V}_{h+1,i}^{k}-\max_{\phi}V_{h+1,i}^{\phi\diamond\pi^k}\right)(s')\geq 0$, and thus $(A)\geq 0$. By uniform concentration, $(B)\leq C\sqrt{SH^2\iota/N_h^k(s,\boldsymbol{a})}=\beta_t$. Putting everything together we have $\overline{Q}_{h,i}^{k}(s,\boldsymbol{a})-\max_{\phi}Q_{h,i}^{\phi\diamond\pi^k}(s,\boldsymbol{a})\geq 0$. The second inequality can be proved similarly.

Now assume inequality (B.42) holds for the h-th step, by the definition of V-functions and CE,

$$\overline{V}_{h,i}^{k}(s)=\mathbb{D}_{\pi^k}\overline{Q}_{h,i}^{k}(s)=\max_{\phi}\mathbb{D}_{\phi\diamond\pi^k}\overline{Q}_{h,i}^{k}(s).$$

By Bellman equation,

$$\max_{\phi}V_{h,i}^{\phi\diamond\pi^k}(s)=\max_{\phi}\mathbb{D}_{\phi\diamond\pi^k}\max_{\phi'}Q_{h,i}^{\phi'\diamond\pi^k}(s).$$

Since by induction hypothesis, for any $(s,\boldsymbol{a})$, $\overline{Q}_{h,i}^{k}(s,\boldsymbol{a})\geq\max_{\phi}Q_{h,i}^{\phi\diamond\pi^k}(s,\boldsymbol{a})$. As a result, we also have $\overline{V}_{h,i}^{k}(s)\geq\max_{\phi}V_{h,i}^{\phi\diamond\pi^k}(s)$, which is exactly inequality (B.35) for the h-th step. The second inequality can be proved similarly. □

B.3.2 Proof of Theorem 21

In this section, we prove each theorem for the single reward function case, i.e., $N=1$. The proof for the case of multiple reward functions ($N>1$) simply follows from taking a union bound, that is, replacing the failure probability p by Np.

NE version Let (μ^k, ν^k) be an arbitrary Nash-equilibrium policy of $\widehat{\mathcal{M}}^k := (\widehat{\mathbb{P}}^k, \widehat{r}^k)$, where $\widehat{\mathbb{P}}^k$ and $\widehat{r}^k$ are our empirical estimate of the transition and the reward at the beginning of the k-th episode in Algorithm 4. Given an arbitrary Nash equilibrium π^k of $\widehat{\mathcal{M}}^k$, we use $\widehat{Q}^k_{h,i}$ and $\widehat{V}^k_{h,i}$ to denote its value functions of the i-th player at the h-th step in $\widehat{\mathcal{M}}^k$.

We prove the following two lemmas, which together imply the conclusion about Nash equilibriums in Theorem 21 as in the proof of Theorem 18.

Lemma 85. *With probability* $1-p$, *for any* $(h, s, \boldsymbol{a}, i)$ *and* $k \in [K]$, *we have*

$$\begin{cases} |\widehat{Q}^k_{h,i}(s,\boldsymbol{a}) - Q^{\pi^k}_{h,i}(s,\boldsymbol{a})| \le \widetilde{Q}^k_h(s,\boldsymbol{a}), \\ |\widehat{V}^k_{h,i}(s) - V^{\pi^k}_{h,i}(s)| \le \widetilde{V}^k_h(s). \end{cases} \tag{B.44}$$

Proof. For each fixed k, we prove this by induction from $h = H+1$ to $h = 1$. For base case, we know at the $(H+1)$-th step,$\widehat{V}^k_{H+1,i} = V^{\pi^k}_{H+1,i} = \widehat{Q}^k_{H+1,i} = Q^{\pi^k}_{H+1,i} = 0$. Now, assume the conclusion holds for the $(h+1)$-th step, for the h-th step, by definition of Q- functions,

$$\begin{aligned}
&\left|\widehat{Q}^k_{h,i}(s,\boldsymbol{a}) - Q^{\pi^k}_{h,i}(s,\boldsymbol{a})\right| \\
\le &\left|\left[\widehat{\mathbb{P}}^k_h \widehat{V}^k_{h+1,i}\right](s,\boldsymbol{a}) - \left[\mathbb{P}_h V^{\pi^k}_{h+1,i}\right](s,\boldsymbol{a})\right| + \left|r_h(s,a) - \widehat{r}^k_h(s,a)\right| \\
\le &\underbrace{\left|\widehat{\mathbb{P}}^k_h\left(\widehat{V}^k_{h+1,i} - V^{\pi^k}_{h+1,i}\right)(s,\boldsymbol{a})\right|}_{(A)} + \underbrace{\left|\left(\widehat{\mathbb{P}}^k_h - \mathbb{P}_h\right)V^{\pi^k}_{h+1,i}(s,\boldsymbol{a})\right| + \left|r_h(s,a) - \widehat{r}^k_h(s,a)\right|}_{(B)}
\end{aligned}$$

By the induction hypothesis,

$$(A) \le \widehat{\mathbb{P}}^k_h\left|\widehat{V}^k_{h+1,i} - V^{\pi^k}_{h+1,i}\right|(s,\boldsymbol{a}) \le (\widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1})(s,\boldsymbol{a}).$$

By uniform concentration [e.g., Lemma 12 in Bai and Jin, 2020], $(B) \le \sqrt{SH^2\iota/N^k_h(s,\boldsymbol{a})} = \beta_t$. Putting everything together we have

$$\left|Q^{\pi^k}_{h,i}(s,\boldsymbol{a}) - \widehat{Q}^k_{h,i}(s,\boldsymbol{a})\right| \le \min\left\{(\widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1})(s,\boldsymbol{a}) + \beta_t, H\right\} = \widetilde{Q}^k_h(s,\boldsymbol{a}),$$

which proves the first inequality in (B.44). The inequality for V functions follows directly by noting that the value functions are computed using the same policy π^k. $\square$

Lemma 86. *With probability* $1-p$*, for any* $(h, s, \boldsymbol{a}, i, k)$*, we have*

$$\begin{cases} |\widehat{Q}^k_{h,i}(s,\boldsymbol{a}) - Q^{\pi^k_{-i},\dagger}_{h,i}(s,\boldsymbol{a})| \le \widetilde{Q}^k_h(s,\boldsymbol{a}), \\ |\widehat{V}^k_{h,i}(s) - V^{\pi^k_{-i},\dagger}_{h,i}(s)| \le \widetilde{V}^k_h(s). \end{cases} \tag{B.45}$$

Proof. For each fixed k, we prove this by induction from $h = H+1$ to $h = 1$. For the base case, we know at the $(H+1)$-th step,$\widehat{V}^k_{H+1,i} = V^{\pi^k_{-i},\dagger}_{H+1,i} = \widehat{Q}^k_{H+1,i} = Q^{\pi^k_{-i},\dagger}_{H+1,i} = 0$. Now, assume the conclusion holds for the $(h+1)$-th step, for the h-th step, by definition of the Q functions,

$$\begin{aligned} &\left|\widehat{Q}^k_{h,i}\left(s,\boldsymbol{a}\right) - Q^{\pi^k_{-i},\dagger}_{h,i}\left(s,\boldsymbol{a}\right)\right| \\ =&\left|\left[\widehat{\mathbb{P}}^k_h \widehat{V}^k_{h+1,i}\right]\left(s,\boldsymbol{a}\right) - \left[\mathbb{P}_h V^{\pi^k_{-i},\dagger}_{h+1,i}\right]\left(s,\boldsymbol{a}\right)\right| + \left|r_h(s,a) - \widehat{r}^k_h(s,a)\right| \\ \le&\underbrace{\left|\widehat{\mathbb{P}}^k_h\left(\widehat{V}^k_{h+1,i} - V^{\pi^k_{-i},\dagger}_{h+1,i}\right)\left(s,\boldsymbol{a}\right)\right|}_{(A)} + \underbrace{\left|\left(\widehat{\mathbb{P}}^k_h - \mathbb{P}_h\right)V^{\pi^k_{-i},\dagger}_{h+1,i}\left(s,\boldsymbol{a}\right)\right| + \left|r_h(s,a) - \widehat{r}^k_h(s,a)\right|}_{(B)} \end{aligned}$$

By the induction hypothesis,

$$(A) \le \widehat{\mathbb{P}}^k_h \left|\widehat{V}^k_{h+1,i} - V^{\pi^k_{-i},\dagger}_{h+1,i}\right|(s,\boldsymbol{a}) \le (\widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1})(s,\boldsymbol{a}).$$

By uniform concentration, $(B) \le \sqrt{SH^2\iota/N^k_h(s,\boldsymbol{a})} = \beta_t$. Putting everything together we have

$$\left|Q^{\pi^k_{-i},\dagger}_{h,i}\left(s,\boldsymbol{a}\right) - \widehat{Q}^k_{h,i}\left(s,\boldsymbol{a}\right)\right| \le \min\left\{(\widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1})(s,\boldsymbol{a}) + \beta_t, H\right\} = \widetilde{Q}^k_h(s,\boldsymbol{a}),$$

which proves the first inequality in (B.45). It remains to show the inequality for V-functions also hold in the h-th step.

Since π^k is a Nash-equilibrium policy, we have

$$\widehat{V}^k_{h,i}\left(s\right) = \max_{\mu} \mathbb{D}_{\mu\times\pi^k_{-i}} \widehat{Q}^k_{h,i}\left(s\right).$$

By Bellman equation,

$$V_{h,i}^{\pi_{-i}^k,\dagger}(s) = \max_{\mu} \mathbb{D}_{\mu\times\pi_{-i}^k} Q_{h,i}^{\pi_{-i}^k,\dagger}(s) .$$

Combining the two equations above, and utilizing the bound we just proved for Q functions, we obtain

$$\begin{aligned} \left|\widehat{V}_{h,i}^k(s) - V_{h,i}^{\pi_{-i}^k,\dagger}(s)\right| &\le \left|\max_{\mu} \mathbb{D}_{\mu\times\pi_{-i}^k} \widehat{Q}_{h,i}^k(s) - \max_{\mu} \mathbb{D}_{\mu\times\pi_{-i}^k} Q_{h,i}^{\pi_{-i}^k,\dagger}(s)\right| \\ &\le \max_{\boldsymbol{a}} \widetilde{Q}_h^k(s,\boldsymbol{a}) = \widetilde{V}_h^k(s), \end{aligned}$$

which completes the whole proof. □

CCE version The proof is almost the same as that for Nash equilibriums. We will reuse Lemma 85 and prove an analogue of Lemma 86. The conclusion for CCEs will follow directly by combining the two lemmas as in the proof of Theorem 18.

Lemma 87. *With probability* $1-p$, *for any* $(h,s,\boldsymbol{a},i)$ *and* $k\in[K]$, *we have*

$$\begin{cases} Q_{h,i}^{\pi_{-i}^k,\dagger}(s,\boldsymbol{a}) - \widehat{Q}_{h,i}^k(s,\boldsymbol{a}) \le \widetilde{Q}_h^k(s,\boldsymbol{a}), \\ V_{h,i}^{\pi_{-i}^k,\dagger}(s) - \widehat{V}_{h,i}^k(s) \le \widetilde{V}_h^k(s). \end{cases} \tag{B.46}$$

Proof. For each fixed k, we prove this by induction from $h=H+1$ to $h=1$. For base case, we know at the $(H+1)$-th step,$\widehat{V}_{H+1,i}^k = V_{H+1,i}^{\pi_{-i}^k,\dagger} = \widehat{Q}_{H+1,i}^k = Q_{H+1,i}^{\pi_{-i}^k,\dagger} = 0$. Now, assume the conclusion holds for the $(h+1)$-th step, for the h-th step, by definition of Q -functions,

$$\begin{aligned} & Q_{h,i}^{\pi_{-i}^k,\dagger}(s,\boldsymbol{a}) - \widehat{Q}_{h,i}^k(s,\boldsymbol{a}) \\ \le & \left[\mathbb{P}_h V_{h+1,i}^{\pi_{-i}^k,\dagger}\right](s,\boldsymbol{a}) - \left[\widehat{\mathbb{P}}_h^k \widehat{V}_{h+1,i}^k\right](s,\boldsymbol{a}) + \left|r_h(s,a) - \widehat{r}_h^k(s,a)\right| \\ \le & \underbrace{\widehat{\mathbb{P}}_h^k\left(V_{h+1,i}^{\pi_{-i}^k,\dagger} - \widehat{V}_{h+1,i}^k\right)(s,\boldsymbol{a})}_{(A)} + \underbrace{\left(\mathbb{P}_h - \widehat{\mathbb{P}}_h^k\right) V_{h+1,i}^{\pi_{-i}^k,\dagger}(s,\boldsymbol{a}) + \left|r_h(s,a) - \widehat{r}_h^k(s,a)\right|}_{(B)}. \end{aligned}$$

By the induction hypothesis, $(A) \le (\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s,\boldsymbol{a})$.

By uniform concentration, $(B) \leq \sqrt{SH^2\iota/N_h^k(s,\boldsymbol{a})} = \beta_t$. Putting everything together we have

$$Q_{h,i}^{\pi_{-i}^k,\dagger}(s,\boldsymbol{a}) - \widehat{Q}_{h,i}^k(s,\boldsymbol{a}) \leq \min\left\{(\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s,\boldsymbol{a}) + \beta_t, H\right\} = \widetilde{Q}_h^k(s,\boldsymbol{a}),$$

which proves the first inequality in (B.46). It remains to show the inequality for V-functions also hold in the h-th step.

Since π^k is a CCE, we have

$$\widehat{V}_{h,i}^k(s) \geq \max_\mu \mathbb{D}_{\mu\times\pi_{-i}^k} \widehat{Q}_{h,i}^k(s).$$

Observe that $V_{h,i}^{\pi_{-i}^k,\dagger}$ obeys the Bellman optimality equation, so we have

$$V_{h,i}^{\pi_{-i}^k,\dagger}(s) = \max_\mu \mathbb{D}_{\mu\times\pi_{-i}^k} Q_{h,i}^{\pi_{-i}^k,\dagger}(s).$$

Combining the two equations above, and utilizing the bound we just proved for Q-functions, we obtain

$$\begin{aligned} V_{h,i}^{\pi_{-i}^k,\dagger}(s) - \widehat{V}_{h,i}^k(s) \leq & \max_\mu \mathbb{D}_{\mu\times\pi_{-i}^k} Q_{h,i}^{\pi_{-i}^k,\dagger}(s) - \max_\mu \mathbb{D}_{\mu\times\pi_{-i}^k} \widehat{Q}_{h,i}^k(s) \\ \leq & \max_{\boldsymbol{a}} \widetilde{Q}_h^k(s,\boldsymbol{a}) = \widetilde{V}_h^k(s), \end{aligned}$$

which completes the whole proof. □

CE version The proof is almost the same as that for Nash equilibriums. We will reuse Lemma 85 and prove an analogue of Lemma 86. The conclusion for CEs will follow directly by combining the two lemmas as in the proof of Theorem 18.

Lemma 88. *With probability $1-p$, for any $(h,s,\boldsymbol{a},i)$, $k\in[K]$ and strategy modification ϕ for player i, we have*

$$\begin{cases} Q_{h,i}^{\phi\diamond\pi^k}(s,\boldsymbol{a}) - \widehat{Q}_{h,i}^k(s,\boldsymbol{a}) \leq \widetilde{Q}_h^k(s,\boldsymbol{a}), \\ V_{h,i}^{\phi\diamond\pi^k}(s) - \widehat{V}_{h,i}^k(s) \leq \widetilde{V}_h^k(s). \end{cases} \tag{B.47}$$

Proof. For each fixed k, we prove this by induction from $h = H+1$ to $h = 1$. For the base case, we know at the $(H+1)$-th step, $\widehat{V}^k_{H+1,i} = V^{\phi\diamond\pi^k}_{H+1,i} = \widehat{Q}^k_{H+1,i} = Q^{\phi\diamond\pi^k}_{H+1,i} = 0$. Now, assume the conclusion holds for the $(h+1)$-th step, for the h-th step, following exactly the same argument as Lemma 87, we can show

$$Q^{\phi\diamond\pi^k}_{h,i}(s,\boldsymbol{a}) - \widehat{Q}^k_{h,i}(s,\boldsymbol{a}) \le \min\left\{(\widehat{\mathbb{P}}^k_h \widetilde{V}^k_{h+1})(s,\boldsymbol{a}) + \beta_t, H\right\} = \widetilde{Q}^k_h(s,\boldsymbol{a}),$$

which proves the first inequality in (B.47). It remains to show the inequality for V-functions also hold in the h-th step.

Since π^k is a CE, we have

$$\widehat{V}^k_{h,i}(s) = \max_{\tilde{\phi}_{h,s}} \mathbb{D}_{\tilde{\phi}_{h,s}\diamond\pi^k} \widehat{Q}^k_{h,i}(s),$$

where the maximum is take over all possible functions from $\mathcal{A}_i$ to itself.

Observe that $V^{\phi\diamond\pi^k}_{h,i}$ obeys the Bellman optimality equation, so we have

$$V^{\phi\diamond\pi^k}_{h,i}(s) = \max_{\tilde{\phi}_{h,s}} \mathbb{D}_{\tilde{\phi}_{h,s}\diamond\pi^k} Q^{\phi\diamond\pi^k}_{h,i}(s).$$

Combining the two equations above, and utilizing the bound we just proved for Q-functions, we obtain

$$\begin{aligned} V^{\phi\diamond\pi^k}_{h,i}(s) - \widehat{V}^k_{h,i}(s) &= \max_{\tilde{\phi}_{h,s}} \mathbb{D}_{\tilde{\phi}_{h,s}\diamond\pi^k} Q^{\phi\diamond\pi^k}_{h,i}(s) - \max_{\tilde{\phi}_{h,s}} \mathbb{D}_{\tilde{\phi}_{h,s}\diamond\pi^k} \widehat{Q}^k_{h,i}(s) \\ &\le \max_{\boldsymbol{a}} \widetilde{Q}^k_h(s,\boldsymbol{a}) = \widetilde{V}^k_h(s), \end{aligned}$$

which completes the whole proof. □

Appendix C

Proof for Chapter 4

C.1 Notations and Basic Lemmas

C.1.1 Notations

In this subsection, we introduce some notations that will be frequently used in appendixes. Recall that we use V^k, N^k, π^k to denote the value, counter and policy maintained by V-learning algorithm at *the beginning* of the episode k.

We also introduce a new policy $\widehat{\pi}_h^k$ for a single agent (defined by its execution in Algorithm 18), which can be viewed as a part of the output policy in Algorithm 7. The definition of $\widehat{\pi}_h^k$ is very similar to $\widehat{\pi}$ except two differences: (1) $\widehat{\pi}_h^k$ is a policy for step $h, \ldots, H$ while $\widehat{\pi}$ is a policy for step $1, \ldots, H$; (2) in $\widehat{\pi}$ the initial value of k is sampled uniformly at random from $[K]$ at the very beginning while in $\widehat{\pi}_h^k$ the initial value of k is given.

We remark that $\widehat{\pi}_h^k$ is a non-Markov policy that does not depends on history before to the $ht, (h)$ step. In symbol, we can express this class of policy as $\pi_j := \{\pi_{j,h'} : \Omega \times (\mathcal{S} \times \mathcal{A})^{h'-h} \times \mathcal{S} \to \mathcal{A}_j\}_{h'=h}^{H}$. We call this class of policy the *policy starting from the $ht, (h)$ step*, and denote it as Π_h. Similar to Section 2.1.1, we can also define joint policy $\pi = \pi_1 \odot \ldots \odot \pi_m$ and product policy $\pi = \pi_1 \times \ldots \times \pi_m$ for policies in Π_h. We

Algorithm 18 Executing Policy $\widehat{\pi}_h^k$

```
for step $h' = h, h+1, \ldots, H$ do
    observe $s_{h'}$, and set $t \leftarrow N_{h'}^k(s_{h'})$.
    set $k \leftarrow k_{h'}^i(s_{h'})$, where $i \in [t]$ is sampled with probability $\alpha_t^i$.
    take action $a_{h'} \sim \pi_{h'}^k(\cdot|s_{h'})$.
```

can also define value $V_h^\pi(s)$ for joint policy $\pi \in \Pi_h$ as

$$V_{i,h}^{\pi}(s) := \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{i,h'}(s_{h'}, \boldsymbol{a}_{h'}) \middle| s_h = s \right].$$

This allows us to define the corresponding best response of π_{-i} as the maximizer of $\max_{\pi_i' \in \Pi_h} V_{i,h}^{\pi_i' \times \pi_{-i}}(s)$. We also denote this maximum value as $V_{i,h}^{\dagger, \pi_{-i}}(s)$. We define the strategy modification for policies starting from the $ht, (h)$ step as $\phi_i := \{\phi_{i,h'} : (\mathcal{S} \times \mathcal{A})^{h'-h} \times \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i\}_{h'=h}^{H}$, and denote the set of such strategy modification as Φ_h.

Finally, for simplicity of notation, we define two operators $\mathbb{P}$ and $\mathbb{D}$ as follows:

$$\begin{cases} \mathbb{P}_h[V](s, \boldsymbol{a}) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,\boldsymbol{a})}[V(s')], \\ \mathbb{D}_\pi[Q](s) = \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|s)}[Q(s, \boldsymbol{a})], \end{cases} \tag{C.1}$$

for any value function V, Q and any one-step Markov policy π.

C.1.2 Basic lemmas

Here we present some basic lemmas that will be used in the proofs of different theorems. We start by introducing some useful properties of sequence $\{\alpha_t^i\}$ defined in (4.2).

Lemma 89. *([Jin et al., 2018, Lemma 4.1],[Tian et al., 2021, Lemma 2]) The following properties hold for α_t^i:*

1. $\frac{1}{\sqrt{t}} \le \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} \le \frac{2}{\sqrt{t}}$ *and* $\frac{1}{t} \le \sum_{i=1}^t \frac{\alpha_t^i}{i} \le \frac{2}{t}$ *for every* $t \ge 1$.

2. $\max_{i \in [t]} \alpha_t^i \le \frac{2H}{t}$ *and* $\sum_{i=1}^t (\alpha_t^i)^2 \le \frac{2H}{t}$ *for every* $t \ge 1$.

3. $\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H}$ *for every* $i \geq 1$.

Finally, we have the following lemma which express the $\tilde{V}$ maintained in V-learning in the form of weighted sum of earlier updates.

Lemma 90. *Consider an arbitrary fixed* (s, h, k) *tuple. Let* $t = N_h^k(s)$ *denote the number of times* s *is visited at step* h *at the beginning of episode* k, *and suppose* s *was previously visited at episodes* $k^1, \ldots, k^t < k$ *at the* h*-th step. Then the two V-values* $\tilde{V}$ *and* V *in Algorithm 5 satisfy the following equation:*

$$\tilde{V}_{j,h}^k(s) = \alpha_t^0(H - h + 1) + \sum_{i=1}^{t} \alpha_t^i \left[r_{j,h}(s, \boldsymbol{a}_h^{k^i}) + V_{j,h+1}^{k^i}(s_{h+1}^{k^i}) + \beta_{j,i} \right], \quad j \in [m]. \quad \text{(C.2)}$$

Proof of Lemma 90. The proof follows directly from the update rule in Line 7 Algorithm 5. Note that α_t^0 is equal to zero for any $t > 1$ and equal to one for $t = 0$. □

C.2 Proofs for Computing CCE in General-sum MGs

In this section, we give complete proof of Theorem 24. To avoid repeatedly state the condition of Theorem 24 in each lemma, we will

- use condition of the adversarial bandit sub-procudure (Assumption 1) and
- set the bonus $\{\beta_{j,t}\}_{t=1}^K$ of the $jt, (h)$ player so that $\sum_{i=1}^t \alpha_t^i \beta_{j,i} = \Theta(H\xi(A_j, t, \iota) + \sqrt{H^3 \iota / t})$ for any $t \in [K]$.

throughout the whole section.

Proof overview. To prove Theorem 24, we need to bound the sum

$$\sum_{k=1}^{K} \max_j [V_{j,1}^{\dagger, \widehat{\pi}_{-j,1}^k} - V_{j,1}^{\widehat{\pi}_1^k}](s_1).$$

By introducing a pessimistic estimation $\underline{V}$ as in Equation (C.3) and Equation (C.4), we first upper and lower bound the value functions by $V_{j,h}^k(s) \geq V_{j,h}^{\dagger, \widehat{\pi}_{-j,h}^k}(s)$ (Lemma 92) and $\underline{V}^k_{j,h}(s) \le V^{\widehat{\pi}^k_h}_{j,h}(s)$ (Lemma 93). As a result, it remains to bound $\sum_{k=1}^K \max_j (V^k_{j,1} - \underline{V}^k_{j,1})(s_1)$ which we handle at the end of this section.

The following Lemma is a direct consequence of Assumption 1, which will play an important role in our later analysis.

Lemma 91. *Under Assumption 1, the following event is true with probability at least $1-\delta$: for any $(s,h,k) \in \mathcal{S} \times [H] \times [K]$, let $t = N^k_h(s)$ and suppose s was previously visited at episodes $k^1, \dots, k^t < k$ at the h-th step, then for all $j \in [m]$*

$$\max_\mu \sum_{i=1}^t \alpha^i_t \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1} \right)(s) - \sum_{i=1}^t \alpha^i_t \mathbb{D}_{\pi^{k^i}_h} \left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1} \right)(s) \le H\xi(A_j, t, \iota),$$

where $\iota = \log(mHSAK/\delta)$.

Proof of Lemma 91. By Assumption 1 and the adversarial bandit update step in Algorithm 5, we have that with probability at least $1-\delta$, for any $(s,h,k,j) \in \mathcal{S} \times [H] \times [K] \times [m]$,

$$\max_\mu \sum_{i=1}^t \alpha^i_t \mathbb{D}_{\pi^{k^i}_h} \left(\frac{H - r_{j,h} - \mathbb{P}_h V^{k^i}_{j,h+1}}{H} \right)(s) - \sum_{i=1}^t \alpha^i_t \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(\frac{H - r_{j,h} - \mathbb{P}_h V^{k^i}_{j,h+1}}{H} \right)(s) \le \xi(A_j, t, \iota),$$

which implies the desired result by simple algebraic transformation. □

Then we show V is actually an optimistic estimation of the value function of player j'th best response to the output policy.

Lemma 92 (Optimism)**.** *For any $\delta \in (0,1]$, with probability at least $1-\delta$, for any $(s,h,k,j) \in \mathcal{S} \times [H] \times [K] \times [m]$, $V^k_{j,h}(s) \ge V^{\dagger, \widehat{\pi}^k_{-j,h}}_{j,h}(s)$.*

Proof of Lemma 92. We prove by backward induction. The claim is satisfied for $h = H+1$ because by definition they are both zero. Suppose it is true for $h+1$ and consider a fixed state s. It suffices to show $\tilde{V}^k_{j,h}(s) \ge V^{\dagger, \widehat{\pi}^k_{-j,h}}_{j,h}(s)$ because $V^k_{j,h}(s) = \min\{\tilde{V}^k_{j,h}(s), H-h+1\}$. Let $t = N^k_h(s)$ and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the h-th step. Then using Lemma 90,

$$
\begin{aligned}
&\tilde{V}^k_{j,h}(s)\\
&= \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left[r_{j,h}(s, \boldsymbol{a}_h^{k^i}) + V^{k^i}_{j,h+1}(s^{k^i}_{h+1}) + \beta_{j,i} \right]\\
&\overset{(i)}{\geq} \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1} \right)(s) + \sum_{i=1}^t \alpha_t^i \beta_{j,i} - \mathcal{O}\left(\sqrt{\frac{H^3 \iota}{t}} \right)\\
&\overset{(ii)}{\geq} \max_\mu \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1} \right)(s) + \sum_{i=1}^t \alpha_t^i \beta_{j,i} - \mathcal{O}\left(\sqrt{\frac{H^3 \iota}{t}} \right) - H\xi(A_j, t, \iota)\\
&\overset{(iii)}{\geq} \max_\mu \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1} \right)(s)\\
&\overset{(iv)}{\geq} \max_\mu \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(r_{j,h} + \mathbb{P}_h V^{\dagger, \widehat{\pi}^{k^i}_{-j,h+1}}_{j,h+1} \right)(s) \overset{(v)}{\geq} V^{\dagger, \widehat{\pi}^k_{-j,h}}_{j,h}(s)
\end{aligned}
$$

where (i) is by martingale concentration and Lemma 2, (ii) is by Lemma 91, (iii) is by the definition of $\beta_{j,i}$, and (iv) is by induction hypothesis.

Finally, we remark that (v) is not directly from Bellman equation since $\widehat{\pi}^k_{-j,h}$ is non-Markov policy, and the best reponse of a non-Markov policy is not necessary a Markov policy. We prove (v) as follows. Recall definitions for policies in Π_h as in Appendix C.1. By the definition, we have

$$
\begin{aligned}
V^{\dagger, \widehat{\pi}^k_{-j,h}}_{j,h}(s) =& \max_{\mu \in \Pi_h} V^{\mu \times \widehat{\pi}^k_{-j,h}}_{j,h}(s)\\
&\overset{(a)}{=} \max_{\mu_h} \max_{\mu_{(h+1):H}} \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\boldsymbol{a} \sim \mu_h \times \pi^{k^i}_{-j,h}} \left(r_{j,h}(s, \boldsymbol{a}) + \mathbb{E}_{s'} V^{\mu_{(h+1):H}, \widehat{\pi}^{k^i}_{-j,h+1}}_{j,h+1}(s, \boldsymbol{a}, s') \right)\\
&\overset{(b)}{\leq} \max_{\mu_h} \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\boldsymbol{a} \sim \mu_h \times \pi^{k^i}_{-j,h}} \left(r_{j,h}(s, \boldsymbol{a}) + \mathbb{E}_{s'} \max_{\mu_{(h+1):H}} V^{\mu_{(h+1):H}, \widehat{\pi}^{k^i}_{-j,h+1}}_{j,h+1}(s, \boldsymbol{a}, s') \right)\\
&\overset{(c)}{=} \max_{\mu_h} \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\boldsymbol{a} \sim \mu_h \times \pi^{k^i}_{-j,h}} \left(r_{j,h}(s, \boldsymbol{a}) + \mathbb{E}_{s'} V^{\dagger, \widehat{\pi}^{k^i}_{-j,h+1}}_{j,h+1}(s') \right)\\
&= \max_\mu \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu \times \pi^{k^i}_{-j,h}} \left(r_{j,h} + \mathbb{P}_h V^{\dagger, \widehat{\pi}^{k^i}_{-j,h+1}}_{j,h+1} \right)(s)
\end{aligned}
$$

where $V^{\pi}_{j,h+1}(s,\boldsymbol{a},s')$ for policy $\pi \in \Pi_h$ is defined as:

$$V^{\pi}_{j,h+1}(s,\boldsymbol{a},s') := \mathbb{E}_{\pi}\left[\textstyle\sum_{h'=h+1}^{H} r_{j,h'}(s_{h'},\boldsymbol{a}_{h'})\middle| s_h = s, \boldsymbol{a}_h = \boldsymbol{a}, s_{h+1} = s'\right].$$

Step (a) uses the relation between $\widehat{\pi}^{k}_{-j,h}$ and $\{\widehat{\pi}^{k^i}_{-j,h+1}\}_i$. Step (b) pushes max inside summation and expectation. Step (c) is because the Markov nature of Markov game and that $\{\widehat{\pi}^{k^i}_{-j,h+1}\}_i$ are policies that does not depend on history at step h, we know the maximization over $\mu_{(h+1):H}$ is achieved at policies in Π_{h+1}. This finishes the proof. □

To proceed with the analysis, we need to introduce two pessimistic V-estimations $\underset{\sim}{V}$ and $\underline{V}$ that are defined similarly as $\tilde{V}$ and V. Formally, let $t = N^k_h(s)$ denote the number of times s is visited at step h at the beginning of episode k, and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the h-th step. Then

$$\underset{\sim}{V}^{k}_{j,h}(s) = \sum_{i=1}^{t} \alpha^i_t \left[r_{j,h}(s, \boldsymbol{a}^{k^i}_h) + \underline{V}^{k^i}_{j,h+1}(s^{k^i}_{h+1}) - \beta_{j,i}\right], \tag{C.3}$$

$$\underline{V}^{k}_{j,h}(s) = \max\{0, \underset{\sim}{V}^{k}_{j,h}(s)\}, \tag{C.4}$$

for any player $j \in [m]$ and $k \in [K]$. We also set $\underline{V}^{k}_{j,H+1}(s) = 0$ for any k, j and s. We emphasize that $\underset{\sim}{V}$ and $\underline{V}$ are defined only for the purpose of analysis. Neither do they influence the decision made by each agent, nor do the agents need to maintain these quantities when running V-learning.

Equipped with the lower estimations, we are ready to lower bound $V^{\widehat{\pi}^k_h}_{j,h}$.

Lemma 93 (Pessimism)**.** *For any $\delta \in (0,1]$, with probability at least $1-\delta$, the following holds for any $(s,h,k,j) \in \mathcal{S} \times [H] \times [K] \times [m]$, $\underline{V}^{k}_{j,h}(s) \le V^{\widehat{\pi}^k_h}_{j,h}(s)$.*

Proof of Lemma 93. We prove by backward induction. The claim is satisfied for $h = H+1$ because by definition they are both zero. Suppose it is true for $h+1$ and consider a fixed state s. It suffices to show $\underset{\sim}{V}^{k}_{j,h}(s) \le V^{\widehat{\pi}^k_h}_{j,h}(s)$ because $\underline{V}^{k}_{j,h}(s) = \max\{\underset{\sim}{V}^{k}_{j,h}(s), 0\}$. Let $t = N^k_h(s)$ and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the

h-th step. Then by Equation C.3,

$$
\begin{aligned}
\underline{V}_{j,h}^{k}(s) &= \sum_{i=1}^{t} \alpha_t^i \left[r_{j,h}(s, \boldsymbol{a}_h^{k^i}) + \underline{V}_{j,h+1}^{k^i}(s_{h+1}^{k^i}) - \beta_{j,i} \right] \\
&\overset{(i)}{\leq} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_h + \mathbb{P}_h \underline{V}_{j,h+1}^{k^i} \right)(s) - \sum_{i=1}^{t} \alpha_t^i \beta_{j,i} + \mathcal{O}\left(\sqrt{\frac{H^3 \iota}{t}} \right) \\
&\overset{(ii)}{\leq} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_h + \mathbb{P}_h \underline{V}_{j,h+1}^{k^i} \right)(s) \\
&\overset{(iii)}{\leq} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_h + \mathbb{P}_h V_{j,h+1}^{\widehat{\pi}_h^{k^i}} \right)(s) \\
&= V_{j,h}^{\widehat{\pi}_h^k}(s)
\end{aligned}
$$

where (i) is by martingale concentration, (ii) is by the definition of $\beta_{j,i}$, and (iii) is by induction hypothesis. □

To prove Theorem 24, it remains to bound the gap $\sum_{k=1}^{K} \max_j (V_{j,1}^k - \underline{V}_{j,1}^k)(s_1)$.

Proof of Theorem 24. Consider player j, we define $\delta_{j,h}^k := V_{j,h}^k(s_h^k) - \underline{V}_{j,h}^k(s_h^k) \geq 0$. The non-negativity here is a simple consequence of the update rule and induction. We want to bound $\delta_h^k := \max_j \delta_{j,h}^k$. Let $n_h^k = N_h^k\left(s_h^k\right)$ and suppose s_h^k was previously visited at episodes $k^1, \ldots, k^{n_h^k} < k$ at the h-th step. Now by the update rule of $V_{j,h}^k(s_h^k)$ and $\underline{V}_{j,h}^k(s_h^k)$,

$$
\begin{aligned}
\delta_{j,h}^k =& V_{j,h}^k(s_h^k) - \underline{V}_{j,h}^k(s_h^k) \\
\leq& \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \left[\left(V_{j,h+1}^{k^i} - \underline{V}_{j,h+1}^{k^i} \right) \left(s_{h+1}^{k^i} \right) + 2\beta_{j,i} \right] \\
=& \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{j,h+1}^{k^i} + \mathcal{O}(H\xi(A_j, n_h^k, \iota) + \sqrt{H^3 \iota / n_h^k})
\end{aligned}
$$

where in the last step we have used $\sum_{i=1}^{t} \alpha_t^i \beta_{j,i} = \Theta(H\xi(A_j, t, \iota) + \sqrt{H^3 \iota / t})$.

Now by taking maximum w.r.t. j on both sides and notice $\xi(B, t, \iota)$ is non-decreasing in B, we have

$$\delta_h^k \leq \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} + \mathcal{O}(H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}).$$

Summing the first two terms w.r.t. k and use Lemma 89,

$$\sum_{k=1}^K \alpha_{n_h^k}^0 H = \sum_{k=1}^K H\mathbb{I}\left\{n_h^k = 0\right\} \leq SH,$$

$$\sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} \overset{(i)}{\leq} \sum_{k'=1}^K \delta_{h+1}^{k'} \sum_{i=n_h^{k'}+1}^{\infty} \alpha_i^{n_h^{k'}} \overset{(ii)}{\leq} \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \delta_{h+1}^k.$$

where (i) is by regrouping the summands. Notice for any $k' \in [K]$, $\delta_{h+1}^{k'}$ appears in the summation only with $k > k'$. The first time it appears we have $n_h^k = n_h^{k'} + 1$, the second time $n_h^k = n_h^{k'} + 2$ and so on. Taking the sum first with respect to k' instead of k gives the desired upper bound.

Putting them together,

$$\begin{aligned}\sum_{k=1}^K \delta_h^k =& \sum_{k=1}^K \alpha_{n_h^k}^0 H + \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} + \sum_{k=1}^K \mathcal{O}(H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}) \\ \leq& HS + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \delta_{h+1}^k + \sum_{k=1}^K \mathcal{O}(H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k})\end{aligned}$$

Recursing this argument for $h \in [H]$ gives

$$\sum_{k=1}^K \delta_1^k \leq eSH^2 + e\sum_{h=1}^H \sum_{k=1}^K \mathcal{O}(H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k})$$

By pigeonhole argument,

$$\sum_{k=1}^K (H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}) = \mathcal{O}(1) \sum_s \sum_{n=1}^{N_h^K(s)} \left(H\xi(A, n, \iota) + \sqrt{\frac{H^3\iota}{n}}\right)$$

$$
\begin{aligned}
&\leq \mathcal{O}(1) \sum_{s} \left(H\Xi(A, N_h^K(s), \iota) + \sqrt{H^3 N_h^K(s)\iota} \right) \\
&\leq \mathcal{O}\left(HS\Xi(A, K/S, \iota) + \sqrt{H^3 SK\iota} \right),
\end{aligned}
$$

where in the last step we have used concavity.

Finally take the sum w.r.t. $h \in [H]$ we have

$$
\sum_{k=1}^{K} \max_j [V_{j,1}^k - \underline{V}_{j,1}^k](s_1) \leq \mathcal{O}\left(H^2 S\Xi(A, K/S, \iota) + \sqrt{H^5 SK\iota} \right),
$$

which implies

$$
\max_{j\in[m]} [V_{j,1}^{\dagger,\widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \leq \mathcal{O}((H^2 S/K) \cdot \Xi(A, K/S, \iota) + \sqrt{H^5 S\iota/K}).
$$

□

C.3 Proofs for Computing CE in General-sum MGs

In this section, we give complete proof of Theorem 25. To avoid repeatedly state the condition of Theorem 25 in each lemma, we will

- use condition of the adversarial bandit sub-procudure (Assumption 2) and
- set the bonus $\{\beta_{j,t}\}_{t=1}^K$ of the $jt, (h)$ player so that $\sum_{i=1}^t \alpha_t^i \beta_{j,i} = \Theta(H\xi_{\mathrm{sw}}(A_j, t, \iota) + \sqrt{H^3\iota/t})$ for any $t \in [K]$.

throughout the whole section.

We begin with a swap regret version of Lemma 91.

Lemma 94. *The following event is true with probability at least* $1 - \delta$*: for any* $(s, h, k) \in \mathcal{S} \times [H] \times [K]$*, let* $t = N_h^k(s)$ *and suppose* s *was previously visited at episodes* $k^1, \ldots, k^t < k$ *at the* h*-th step, then for all* $j \in [m]$

$$
\max_{\phi_j} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\phi_j \diamond \pi_{j,h}^{k^i} \times \pi_{-j,h}^{k^i}} \left[r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right](s) - \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right)(s)
$$

$$\leq H\xi_{sw}(A_j, t, \iota),$$

where $\iota = \log(mKHAS/\delta)$.

Proof of Lemma 94. By Assumption 2 and the adversarial bandit update step in Algorithm 5, we have that with probability at least $1-\delta$, for any $(s,h,k,j) \in \mathcal{S} \times [H] \times [K] \times [m]$,

$$\max_{\phi_j} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\phi_j \diamond \pi_{j,h}^{k^i} \times \pi_{-j,h}^{k^i}} \left(\frac{H - r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i}}{H} \right)(s) \\ - \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(\frac{H - r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i}}{H} \right)(s) \leq \xi_{\mathrm{sw}}(A_j, t, \iota).$$

□

We begin with proving V is actually an optimistic estimation of the value function under best response.

Lemma 95 (Optimism)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds for any $(s,h,k,j) \in \mathcal{S} \times [H] \times [K] \times [m]$, $V_{j,h}^k(s) \geq \max_{\phi_j} V_{j,h}^{(\phi_j \diamond \widehat{\pi}_{j,h}^k) \odot \widehat{\pi}_{-j,h}^k}(s)$.*

Proof of Lemma 95. We prove by backward induction. The claim is satisfied for $h = H+1$ because by definition they are both zero. Suppose it is true for $h+1$ and consider a fixed state s. It suffices to show $\tilde{V}_{j,h}^k(s) \geq \max_{\phi_j} V_{j,h}^{(\phi_j \diamond \widehat{\pi}_{j,h}^k) \odot \widehat{\pi}_{-j,h}^k}(s)$ because $V_{j,h}^k(s) = \min\{\tilde{V}_{j,h}^k(s), H-h+1\}$. Let $t = N_h^k(s)$ and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the h-th step. Then using Lemma 90,

$$\begin{aligned} \tilde{V}_{j,h}^k(s) =& \alpha_t^0 (H-h+1) + \sum_{i=1}^{t} \alpha_t^i \left[r_{j,h}(s, \boldsymbol{a}_h^{k^i}) + V_{j,h+1}^{k^i}(s_{h+1}^{k^i}) + \beta_{j,i} \right] \\ \overset{(i)}{\geq} & \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\pi_h^{k^i}} \left(r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right)(s) + \sum_{i=1}^{t} \alpha_t^i \beta_{j,i} - \mathcal{O}\left(\sqrt{\frac{H^3 \iota}{t}} \right) \\ \overset{(ii)}{\geq} & \max_{\phi_j} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{(\phi_j \diamond \pi_{j,h}^{k^i}) \times \pi_{-j,h}^{k^i}} \left(r_h + \mathbb{P}_h V_{j,h+1}^{k^i} \right)(s) + \sum_{i=1}^{t} \alpha_t^i \beta_{j,i} \\ & - \mathcal{O}\left(\sqrt{\frac{H^3 \iota}{t}} \right) - H\xi_{\mathrm{sw}}(A_j, t, \iota) \end{aligned}$$

$$
\begin{aligned}
&\overset{(iii)}{\geq} \max_{\phi_j} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{(\phi_j \diamond \pi_{j,h}^{k^i}) \times \pi_{-j,h}^{k^i}} \left(r_h + \mathbb{P}_h V_{j,h+1}^{k^i} \right)(s) \\
&\overset{(iv)}{\geq} \max_{\phi_j} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{(\phi_j \diamond \pi_{j,h}^{k^i}) \times \pi_{-j,h}^{k^i}} \left(r_h + \mathbb{P}_h \max_{\phi_j'} V_{j,h}^{(\phi_j' \diamond \widehat{\pi}_{j,h+1}^{k^i}) \odot \widehat{\pi}_{-j,h+1}^{k^i}} \right)(s) \\
&\overset{(v)}{\geq} \max_{\phi_j} V_{j,h}^{(\phi_j \diamond \widehat{\pi}_{j,h}^{k}) \odot \widehat{\pi}_{-j,h}^{k}}(s)
\end{aligned}
$$

where (i) is by martingale concentration and Lemma 2, (ii) is by Lemma 94, (iii) is by the definition of $\beta_{j,i}$, and (iv) is by induction hypothesis. Finally, (v) follows from a similar reasoning as in the proof of Lemma 92, which we omit here. □

We still need to lower bound $V_{j,h}^{\widehat{\pi}_h^k}$. To do this, we estimate $\utilde{V}$ and $\underline{V}$ defined by Equation (C.3) and Equation (C.4). Lemma 93 shows these quantities are indeed the lower bounds we need.

To prove Theorem 25, it remains to bound the gap $\sum_{k=1}^{K} \max_j (V_{j,1}^k - \underline{V}_{j,1}^k)(s_1)$. This is actually the same as the concluding part of the proof Theorem 24, except changing Ξ to Ξ_{sw}. For completeness we still keep a shortened version of full proof here.

Proof of Theorem 25. Consider player j, we define $\delta_{j,h}^k := V_{j,h}^k(s_h^k) - \underline{V}_{j,h}^k(s_h^k) \geq 0$. The non-negativity here is a simple consequence of the update rule and induction. We want to bound $\delta_h^k := \max_j \delta_{j,h}^k$. Let $n_h^k = N_h^k\left(s_h^k\right)$ and suppose s_h^k was previously visited at episodes $k^1, \ldots, k^{n_h^k} < k$ at the h-th step. Now by the update rule of $V_{j,h}^k(s_h^k)$ and $\underline{V}_{j,h}^k(s_h^k)$,

$$
\begin{aligned}
\delta_{j,h}^k =& V_{j,h}^k(s_h^k) - \underline{V}_{j,h}^k(s_h^k) \\
\leq& \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \left[\left(V_{j,h+1}^{k^i} - \underline{V}_{j,h+1}^{k^i} \right) \left(s_{h+1}^{k^i} \right) + 2\beta_{j,i} \right] \\
=& \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{j,h+1}^{k^i} + \mathcal{O}(H \xi_{\mathrm{sw}}(A_j, n_h^k, \iota) + \sqrt{H^3 \iota / n_h^k})
\end{aligned}
$$

where in the last step we have used $\sum_{i=1}^{t} \alpha_t^i \beta_{j,i} = \Theta(H \xi_{\mathrm{sw}}(A_j, t, \iota) + \sqrt{H^3 \iota / t})$.

Now by taking maximum w.r.t. j on both sides and notice $\xi_{\mathrm{sw}}(B,t,\iota)$ is non-decreasing in B, we have

$$\delta_h^k \leq \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} + \mathcal{O}(H\xi_{\mathrm{sw}}(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}).$$

Summing the first two terms w.r.t. k,

$$\sum_{k=1}^K \alpha_{n_h^k}^0 H = \sum_{k=1}^K H\mathbb{I}\left\{n_h^k = 0\right\} \leq SH,$$

$$\sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} \overset{(i)}{\leq} \sum_{k'=1}^K \delta_{h+1}^{k'} \sum_{i=n_h^{k'}+1}^{\infty} \alpha_i^{n_h^{k'}} \overset{(ii)}{\leq} \left(1+\frac{1}{H}\right) \sum_{k=1}^K \delta_{h+1}^k.$$

where (i) is by changing the order of summation and (ii) is by Lemma 89. Putting them together,

$$\begin{aligned}
\sum_{k=1}^K \delta_h^k =& \sum_{k=1}^K \alpha_{n_h^k}^0 H + \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} + \sum_{k=1}^K \mathcal{O}(H\xi_{\mathrm{sw}}(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}) \\
\leq& HS + \left(1+\frac{1}{H}\right) \sum_{k=1}^K \delta_{h+1}^k + \sum_{k=1}^K \mathcal{O}(H\xi_{\mathrm{sw}}(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k})
\end{aligned}$$

Recursing this argument for $h \in [H]$ gives

$$\sum_{k=1}^K \delta_1^k \leq eSH^2 + e\sum_{h=1}^H \sum_{k=1}^K \mathcal{O}(H\xi_{\mathrm{sw}}(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k})$$

By pigeonhole argument,

$$\begin{aligned}
\sum_{k=1}^K (H\xi_{\mathrm{sw}}(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}) =& \mathcal{O}\left(1\right) \sum_s \sum_{n=1}^{N_h^K(s)} \left(H\xi_{\mathrm{sw}}(A, n, \iota) + \sqrt{\frac{H^3\iota}{n}}\right) \\
\leq& \mathcal{O}\left(1\right) \sum_s \left(H\Xi_{\mathrm{sw}}(A, N_h^K\left(s\right), \iota) + \sqrt{H^3 N_h^K(s)\iota}\right) \\
\leq& \mathcal{O}\left(HS\Xi_{\mathrm{sw}}(A, K/S, \iota) + \sqrt{H^3SK\iota}\right),
\end{aligned}$$

where in the last step we have used concavity.

Finally take the sum w.r.t. $h \in [H]$ we have

$$\sum_{k=1}^{K} \max_{j} [V_{j,1}^{k} - \underline{V}_{j,1}^{k}](s_1) \leq \mathcal{O}\left(H^2 S \Xi_{\mathrm{sw}}(A, K/S, \iota) + \sqrt{H^5 S K \iota}\right),$$

which implies

$$\max_{j\in[m]} [V_{j,1}^{\dagger, \widehat{\pi}_{-j}}(s_1) - V_{j,1}^{\widehat{\pi}}(s_1)] \leq \mathcal{O}((H^2 S/K) \cdot \Xi_{\mathrm{sw}}(A, K/S, \iota) + \sqrt{H^5 S \iota / K}).$$

□

C.4 Proofs for MDPs and Two-player Zero-sum MGs

In this section, we prove the main theorems for V-learning in the setting of single-agent (MDPs) and two-player zero-sum MGs.

Proof of Theorem 23. To begin with, we notice an equivalent definition of two-player zero-sum MGs is that the reward function satisfies $r_{1,h} = 1 - r_{2,h}$ for all $h \in [H]$. The reason we use this definition instead of the common version $r_{1,h} = -r_{2,h}$ is we want to make it consistent with our assumption that the reward function takes value in $[0, 1]$ for any player. Although this definition does not satisfy the zero-sum condition, its Nash equilibria are the same as those of the zero-sum version because adding a constant to the reward function of player 2 per step will not change the dynamics of the game.

In order to show $\widehat{\pi} = \widehat{\pi}_1 \times \widehat{\pi}_2$ is an approximate Nash policy, it suffices to control

$$\max_{\pi_1} V_{1,1}^{\pi_1, \widehat{\pi}_2}(s_1) - \min_{\pi_2} V_{1,1}^{\widehat{\pi}_1, \pi_2}(s_1).$$

Since $r_{1,h} = 1 - r_{2,h}$ for all $h \in [H]$, with probability at least $1 - \delta$

$$\max_{\pi_1} V_{1,1}^{\pi_1, \widehat{\pi}_2}(s_1) - \min_{\pi_2} V_{1,1}^{\widehat{\pi}_1, \pi_2}(s_1)$$

$$
\begin{aligned}
&= \max_{\pi_1} V_{1,1}^{\pi_1,\widehat{\pi}_2}(s_1) - \left(H - \max_{\pi_2} V_{2,1}^{\widehat{\pi}_1,\pi_2}(s_1) \right) \\
&= \left(\max_{\pi_1} V_{1,1}^{\pi_1,\widehat{\pi}_2}(s_1) - V_{1,1}^{\widehat{\pi}_1 \odot \widehat{\pi}_2}(s_1) \right) + \left(\max_{\pi_2} V_{2,1}^{\widehat{\pi}_1,\pi_2}(s_1) - V_{2,1}^{\widehat{\pi}_1 \odot \widehat{\pi}_2}(s_1) \right) \\
&\le \mathcal{O}((H^2 S/K) \cdot \Xi(A, K/S, \iota) + \sqrt{H^5 S \iota / K}),
\end{aligned}
$$

where the last inequality follows from Theorem 24. The reason we can use Theorem 24 here is the precondition of Theorem 23 is a special case of the precondition of Theorem 24. □

Proof of Theorem 22. Since MDPs is a subclass of two-player zero-sum MGs by simply choosing the action set of the second player to be a singleton, it suffices to only prove Theorem 23, from which the single-agent guarantee, Theorem 22 trivially follows. □

C.5 Proofs for Monotonic V-learning

In this section, we prove Theorem 26. The algorithm is V-learning with monotonic update, and the setting we consider is two-player zero-sum Markov games. As before, we assume $r_{1,h}(s,a) = 1 - r_{2,h}(s,a)$ for all s, a, h. The reason for assuming $r_{1,h}(s,a) = 1 - r_{2,h}(s,a)$ instead of $r_{1,h}(s,a) = -r_{2,h}(s,a)$ can be found in Appendix C.4.

For two player zero-sum MGs, we can define its minimax value function (Nash value function) by the following Bellman equations

$$
\begin{cases}
V_{j,h}^{\star}(s) = \max_{\pi_{j,h}} \min_{\pi_{-j,h}} \mathbb{D}_{\pi_{j,h} \times \pi_{-j,h}}[Q_{j,h}^{\star}](s), \\
Q_{j,h}^{\star}(s, \boldsymbol{a}) = r_{j,h}(s, \boldsymbol{a}) + \mathbb{P}_h[V_{j,h+1}^{\star}](s, \boldsymbol{a}), \\
V_{j,H+1}^{\star}(s) = Q_{j,H+1}^{\star}(s, \boldsymbol{a}) = 0.
\end{cases}
\tag{C.5}
$$

Lemma 96 (Optimism of V-estimates). *With probability at least $1 - \delta$, for any $(s, h, k, j) \in \mathcal{S} \times [H] \times [K] \times [2]$,*

$$
\tilde{V}_{j,h}^{k}(s) \ge V_{j,h}^{k}(s) \ge V_{j,h}^{\dagger, \tilde{\pi}_{-j}}(s) \ge V_{j,h}^{\star}(s),
\tag{C.6}
$$

where $V^{\star}_{j,h}$ is the minimax (Nash) value function defined above.

Proof of Lemma 96. Note that $\tilde{V}^k_{j,h}(s) \geq V^k_{j,h}(s)$ is straightforward by the update rule of V-learning, and $V^{\dagger,\tilde{\pi}_{-j}}_{j,h}(s) \geq V^{\star}_{j,h}(s)$ directly follows from the definition of minimax value function. Therefore, we only need to prove the second inequality. We do this by backward induction.

The claim is true for $h = H+1$. Assume for any s and k, $V^k_{j,h+1}(s) \geq V^{\dagger,\tilde{\pi}_{-j}}_{j,h+1}(s)$. For a fixed $(s,h,k) \in \mathcal{S} \times [H] \times [K]$, let $t = N^k_h(s)$ and suppose s was previously visited in episode $k^1, \ldots, k^t < k$ at the h-th step. By Bellman equation,

$$\begin{aligned}
&V^{\dagger,\tilde{\pi}_{-j}}_{j,h}(s)\\
\leq&\alpha^0_t(H-h+1) + \max_{\mu}\sum_{i=1}^{t}\alpha^i_t \mathbb{D}_{\mu\times\pi^{k^i}_{-j,h}}\left(r_{j,h} + \mathbb{P}_h V^{\dagger,\tilde{\pi}_{-j}}_{j,h+1}\right)(s)\\
\leq&\alpha^0_t(H-h+1) + \max_{\mu}\sum_{i=1}^{t}\alpha^i_t \mathbb{D}_{\mu\times\pi^{k^i}_{-j,h}}\left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1}\right)(s)\\
\leq&\alpha^0_t(H-h+1) + \sum_{i=1}^{t}\alpha^i_t \mathbb{D}_{\pi^{k^i}_h}\left(r_{j,h} + \mathbb{P}_h V^{k^i}_{j,h+1}\right)(s) + H\xi(A_j,t,\iota)\\
\leq&\alpha^0_t(H-h+1) + \sum_{i=1}^{t}\alpha^i_t\left[r_{j,h}(s,\boldsymbol{a}^{k^i}_h) + V^{k^i}_{j,h+1}(s^{k^i}_{h+1})\right] + \mathcal{O}\left(\sqrt{\frac{2H^3\iota}{t}}\right) + H\xi(A_j,t,\iota)
\end{aligned}$$

where the second inequality follows from our induction hypothesis and the monotonicity of V^k, the third inequality follows from Lemma 91, and the last one follows from martingale concentration as well as Lemma 89. By Lemma 90 and the precondition of Theorem 26, we know the RHS is no larger than $\tilde{V}^k_{j,h}(s)$. Note that V^k can be equivalently defined as

$$V^k_{j,h}(s) = \min\{\min_{t\in[k]}\tilde{V}^t_{j,h}(s), H-h+1\},$$

we conclude $V^k_{j,h}(s) \geq V^{\dagger,\tilde{\pi}_{-j}}_{j,h}(s)$ for any $k \in [K]$. □

Now we are ready to prove Theorem 26.

Proof of Theorem 26. By the monotonicity of V and Lemma 96

$$\begin{aligned} V_{1,1}^{\dagger,\tilde{\pi}_2}(s_1) - \min_{\pi_2} V_{1,1}^{\tilde{\pi}_1 \times \pi_2}(s_1) &= V_{1,1}^{\dagger,\tilde{\pi}_2}(s_1) - \left(H - V_{2,1}^{\dagger,\tilde{\pi}_1}(s_1)\right) \\ &\le V_{1,1}^{K}(s_1) + V_{2,1}^{K}(s_1) - H \\ &\le \frac{1}{K}\sum_{k=1}^{K}\left(V_{1,1}^{k}(s_1) + V_{2,1}^{k}(s_1) - H\right) \\ &\le \frac{1}{K}\sum_{k=1}^{K}\left(\tilde{V}_{1,1}^{k}(s_1) + \tilde{V}_{2,1}^{k}(s_1) - H\right), \end{aligned}$$

where the first equality follows from the definition of two-player zero-sum game, i.e., $r_{1,h} = 1 - r_{2,h}$.

Now we can mimic the proof of Theorem 24. Define $\delta_h^k := \tilde{V}_{1,h}^k(s_h^k) + \tilde{V}_{2,h}^k(s_h^k) - (H - h + 1)$. The non-negativity here follows from Lemma 96 as below

$$\begin{aligned} \tilde{V}_{1,h}^k(s_h^k) + \tilde{V}_{2,h}^k(s_h^k) - (H-h+1) \ge& V_{1,h}^\star(s_h^k) + V_{2,h}^\star(s_h^k) - (H-h+1) \\ =&(H-h+1) - (H-h+1) = 0. \end{aligned}$$

Let $n_h^k = N_h^k\left(s_h^k\right)$ and suppose s_h^k was previously visited at episodes $k^1, \ldots, k^{n_h^k} < k$ at the h-th step. By Lemma 90 and the fact that $r_{1,h} = 1 - r_{2,h}$ for all h, we have

$$\begin{aligned} \delta_h^k =& \tilde{V}_{1,h}^k(s_h^k) + \tilde{V}_{2,h}^k(s_h^k) - (H-h+1) \\ =& 2\alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \left[\left(\tilde{V}_{1,h+1}^{k^i} - \tilde{V}_{2,h+1}^{k^i}\right)\left(s_{h+1}^{k^i}\right) - (H-h) + \beta_{1,i} + \beta_{2,i}\right] \\ =& 2\alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k^i} + \mathcal{O}(H\xi(A, n_h^k, \iota) + \sqrt{H^3\iota/n_h^k}) \end{aligned}$$

where in the last step we used $\sum_{i=1}^t \alpha_t^i \beta_{j,i} = \Theta(H\xi(A_j, t, \iota) + \sqrt{H^3\iota/t})$.

The remaining steps follow exactly the same as the proof of Theorem 24. As a result, we obtain

$$V_{1,1}^{\dagger,\tilde{\pi}_2}(s_1) - \min_{\tilde{\pi}_2} V_{1,1}^{\tilde{\pi}_1 \times \tilde{\pi}_2}(s_1) \le \frac{1}{K}\sum_{k=1}^{K}\left(\tilde{V}_{1,1}^{k}(s_1) + \tilde{V}_{2,1}^{k}(s_1) - H\right)$$

Algorithm 19 FTRL for Weighted External Regret (FTRL)

```
1: Initialize: for any $b \in \mathcal{B}$, $\theta_1(b) \leftarrow 1/B$.
2: for episode $t = 1, \ldots, K$ do
3:     Take action $b_t \sim \theta_t(\cdot)$, and observe loss $\tilde{l}_t(b_t)$.
4:     $\widehat{l}_t(b) \leftarrow \tilde{l}_t(b_t)\mathbb{I}\{b_t = b\}/(\theta_t(b) + \gamma_t)$ for all $b \in \mathcal{B}$.
5:     $\theta_{t+1}(b) \propto \exp[-(\eta_t/w_t) \cdot \sum_{i=1}^{t} w_i \widehat{l}_i(b)]$
```

$$\leq \mathcal{O}\left(\frac{H^2 S}{K} \cdot \Xi(A, K/S, \iota) + \sqrt{\frac{H^5 S \iota}{K}}\right),$$

which completes the proof. □

C.6 Adversarial Bandit with Weighted External Regret

In this section, we present a Follow-the-Regularized-Leader (FTRL) style algorithm that achieves low weighted (external) regret for the adversarial bandit problem. Although FTRL is a classial algorithm in the adversarial bandit literature, we did not find a good reference of FTRL with changing step size, weighted regret and high probability bound. For completeness of this work, we provide detailed derivations here.

We present the FTRL algorithm in Algorithm 19. In Corollary 97, we prove that FTRL satisfies the Assumption 1 with good regret bounds. Recall that B is the number of actions, and our normalization condition requires loss $\tilde{l}_t \in [0, 1]^B$ for any t.

Corollary 97. *By choosing hyperparameter $w_t = \alpha_t \left(\prod_{i=2}^{t} (1 - \alpha_i)\right)^{-1}$ and $\eta_t = \gamma_t = \sqrt{\frac{H \log B}{Bt}}$, FTRL (Algorithm 19) satisfies Assumption 1 with*

$$\xi(B, t, \log(1/\delta)) = 10\sqrt{HB \log(B/\delta)/t}, \qquad \Xi(B, t, \log(1/\delta)) = 20\sqrt{HBt \log(B/\delta)}$$

To prove Corollary 97, we show a more general weighted regret guarantee which works for any set of weights $\{w_i\}_{i=1}^{\infty}$ in addition to $\{\alpha_t^i\}_{i=1}^{t}$. In particular, a general weighted regret is defined as

$$\mathcal{R}(t) = \max_{\theta^\star} \sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, l_i \right\rangle \tag{C.7}$$

Theorem 98. *For any $t \le K$, following Algorithm 19, if $\eta_i \le 2\gamma_i$ and η_i is non-increasing for all $i \le t$, let $\iota = \log(B/\delta)$, then with probability $1 - 3\delta$, we have*

$$\mathcal{R}(t) \le \frac{w_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \le t} w_i \iota + B \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2\iota \sum_{i=1}^{t} w_i^2} + \max_{i \le t} w_i \iota / \gamma_t.$$

We postpone the proof of theorem 98 to the end of this section. We first show how to obtain Corollary 97 from Theorem 98.

Proof of Corollary 97. The weights $\{w_t\}_{t=1}^{K}$ we choose satisfy a nice property: for any t we have

$$\frac{w_i}{w_j} = \frac{\alpha_t^i}{\alpha_t^j}.$$

We prove this for $i \le j$ and the other case is similar. By definition,

$$\frac{w_i}{w_j} = \frac{\alpha_i}{\alpha_j} \prod_{k=i+1}^{j} (1 - \alpha_k),$$

and

$$\frac{\alpha_t^i}{\alpha_t^j} = \frac{\alpha_i}{\alpha_j} \prod_{k=i+1}^{j} (1 - \alpha_k).$$

We can easily verify that the RHS are the same.

Define $\tilde{\mathcal{R}}(t) := \max_{\theta \in \Delta_{\mathcal{B}}} \sum_{i=1}^{t} \alpha_t^i [\langle \theta_i, \ell_i \rangle - \langle \theta, \ell_i \rangle]$. By plugging $w_t = \alpha_t \left(\prod_{i=2}^{t} (1 - \alpha_i) \right)^{-1}$ into Theorem 98, and using the property above, we have the regret guarantee

$$\tilde{\mathcal{R}}(t) \le \frac{\alpha_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i \alpha_t^i + \frac{1}{2} \alpha_t \iota + B \sum_{i=1}^{t} \gamma_i \alpha_t^i + \sqrt{2\iota \sum_{i=1}^{t} \left(\alpha_t^i\right)^2} + \alpha_t \iota / \gamma_t.$$

By choosing $\eta_t = \gamma_t = \sqrt{\frac{H \log B}{Bt}}$ and using Lemma 89, we can further upper bound the regret by

$$
\begin{aligned}
\tilde{\mathcal{R}}(t) \leq & \frac{(H+1)\log B}{H+t}\sqrt{\frac{Bt}{H\log B}} + \frac{3}{2}\sqrt{HB\log B}\sum_{i=1}^{t}\frac{\alpha_t^i}{\sqrt{t}} \\
& + \frac{(H+1)\,\iota}{2\,(H+t)} + \sqrt{2\iota\sum_{i=1}^{t}(\alpha_t^i)^2} + \frac{(H+1)\,\iota}{(H+t)}\sqrt{\frac{Bt}{H\log B}} \\
\leq & 2\sqrt{\frac{HB\log B}{t}} + 3\sqrt{\frac{HB\log B}{t}} + \frac{H\iota}{t} + 2\sqrt{\frac{H\iota}{t}} + 2\sqrt{\frac{HB\log B}{t}} \\
\leq & 9\sqrt{HB\iota/t} + H\iota/t.
\end{aligned}
$$

To further simplify the above upper bound, consider two cases:

- If $H\iota/t \leq 1$, $\sqrt{H\iota/t} \geq H\iota/t$ and thus $\tilde{\mathcal{R}}(t) \leq 10\sqrt{HB\iota/t}$.
- If $H\iota/t \geq 1$, $\sqrt{HB\iota/t} \geq 1 \geq \tilde{\mathcal{R}}(t)$ where the last step is by the definition of $\tilde{\mathcal{R}}(t)$. Therefore we have $\tilde{\mathcal{R}}(t) \leq \sqrt{HB\iota/t}$.

Combining the two cases above gives $\tilde{\mathcal{R}}(t) \leq 10\sqrt{HB\iota/t}$.

Finally, we pick $\xi(B, t, \log(1/\delta)) := 10\sqrt{HB\iota/t}$, which is non-decreasing in B. Since $\sum_{t'=1}^{t}\xi(B, t, \log(1/\delta)) \leq 20\sqrt{HBt\iota}$, we choose $\Xi(B, t, \log(1/\delta)) = 20\sqrt{HBt\iota}$, which is concave in t. □

To prove Theorem 98, we first note that the weighted regret (C.7) can be decomposed into three terms

$$
\begin{aligned}
\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, l_i \right\rangle = & \sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, l_i \right\rangle \\
= & \underbrace{\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, \widehat{l}_i \right\rangle}_{(A)} + \underbrace{\sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \widehat{l}_i \right\rangle}_{(B)} + \underbrace{\sum_{i=1}^{t} w_i \left\langle \theta^\star, \widehat{l}_i - l_i \right\rangle}_{(C)}
\end{aligned}
\tag{C.8}
$$

The rest of this section is devoted to bounding three terms above. We begin with the following useful lemma adapted from Lemma 1 in Neu [2015], which is crucial in achieving high probability guarantees.

Lemma 99. *For any sequence of coefficients $c_1, c_2, \dots, c_t$ s.t. $c_i \in [0, 2\gamma_i]^B$ is $\mathcal{F}_i$-measurable, we have with probability $1-\delta$,*

$$\sum_{i=1}^{t} w_i \left\langle c_i, \widehat{l}_i - l_i \right\rangle \le \max_{i \le t} w_i \iota.$$

Proof of Lemma 99. Define $w = \max_{i \le t} w_i$. By definition,

$$\begin{aligned} w_i \widehat{l}_i(b) =& \frac{w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{\theta_i(b) + \gamma_i} \le \frac{w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{\theta_i(b) + \frac{w_i \tilde{l}_i(b) \mathbb{I}\{b_i = b\}}{w} \gamma_i} \\ =& \frac{w}{2\gamma_i} \frac{\frac{2\gamma_i w_i \tilde{l}_i(b) \mathbb{I}\{b_i = b\}}{w\theta_i(b)}}{1 + \frac{\gamma_i w_i \tilde{l}_i(b) \mathbb{I}\{b_i = b\}}{w\theta_i(b)}} \overset{(i)}{\le} \frac{w}{2\gamma_i} \log\left(1 + \frac{2\gamma_i w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{w\theta_i(b)}\right) \end{aligned}$$

where (i) follows from $\frac{z}{1+z/2} \le \log(1+z)$ for all $z \ge 0$.

Defining the sum

$$\widehat{S}_i = \frac{w_i}{w} \left\langle c_i, \widehat{l}_i \right\rangle, \quad S_i = \frac{w_i}{w} \left\langle c_i, l_i \right\rangle,$$

we have

$$\begin{aligned} \mathbb{E}_i\left[\exp\left(\widehat{S}_i\right)\right] &\le \mathbb{E}_i\left[\exp\left(\sum_b \frac{c_i(b)}{2\gamma_i} \log\left(1 + \frac{2\gamma_i w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{w\theta_i(b)}\right)\right)\right] \\ &\overset{(i)}{\le} \mathbb{E}_i\left[\prod_b \left(1 + \frac{c_i(b)\, w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{w\theta_i(b)}\right)\right] \\ &= \mathbb{E}_i\left[1 + \sum_b \frac{c_i(b)\, w_i \tilde{l}_i(b)\, \mathbb{I}\{b_i = b\}}{w\theta_i(b)}\right] \\ &= 1 + S_i \le \exp(S_i) \end{aligned}$$

where (i) follows from $z_1 \log(1+z_2) \le \log(1+z_1 z_2)$ for any $0 \le z_1 \ge 1$ and $z_2 \ge -1$. Note that here we are using the condition $c_i(b) \le 2\gamma_i$ for all $b \in [B]$.

Equipped with the above bound, we can now prove the concentration result.

$$\mathbb{P}\left[\sum_{i=1}^{t} \left(\widehat{S}_i - S_i\right) \ge \iota\right] = \mathbb{P}\left[\exp\left[\sum_{i=1}^{t} \left(\widehat{S}_i - S_i\right)\right] \ge \frac{B}{\delta}\right]$$

$$
\begin{aligned}
&\le \frac{\delta}{B}\mathbb{E}_t\left[\exp\left[\sum_{i=1}^{t}\left(\widehat{S}_i - S_i\right)\right]\right]\\
&\le \frac{\delta}{B}\mathbb{E}_{t-1}\left[\exp\left[\sum_{i=1}^{t-1}\left(\widehat{S}_i - S_i\right)\right] E_t\left[\exp\left(\widehat{S}_t - S_t\right)\right]\right]\\
&\le \frac{\delta}{B}\mathbb{E}_{t-1}\left[\exp\left[\sum_{i=1}^{t-1}\left(\widehat{S}_i - S_i\right)\right]\right]\\
&\le \cdots \le \frac{\delta}{B}.
\end{aligned}
$$

We conclude the proof by taking a union bound. □

With Lemma 99, we can bound the three terms (A),(B) and (C) in (C.8) separately as below.

Lemma 100. *For any $t \in [K]$, suppose $\eta_i \le 2\gamma_i$ for all $i \le t$. Then with probability at least $1-\delta$, for any $\theta^\star \in \Delta^B$,*

$$
\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, \widehat{l}_i \right\rangle \le \frac{w_t \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t}\eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota.
$$

Proof of Lemma 100. We use the standard analysis of FTRL with changing step size, see for example Exercise 28.13 in Lattimore and Szepesvári [2020]. Notice the essential step size is η_t/w_t,

$$
\begin{aligned}
\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^\star, \widehat{l}_i \right\rangle &\le \frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\eta_i w_i \left\langle \theta_i, \widehat{l}_i^2 \right\rangle\\
&\le \frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b\in\mathcal{B}}\eta_i w_i \widehat{l}_i\left(b\right)\\
&\overset{(i)}{\le} \frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b\in\mathcal{B}}\eta_i w_i l_i\left(b\right) + \frac{1}{2}\max_{i\le t} w_i \iota\\
&\le \frac{w_t \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t}\eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota
\end{aligned}
$$

where (i) is by using Lemma 99 with $c_i(b) = \eta_i$ for any b. The any-time guarantee is justifed by taking union bound. □

Lemma 101. *For any $t \in [K]$, with probability $1-\delta$,*

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \widehat{l}_i \right\rangle \le B \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2\iota \sum_{i=1}^{t} w_i^2}.$$

Proof of Lemma 101. We further decopose it into

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \widehat{l}_i \right\rangle = \sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \mathbb{E}_i \widehat{l}_i \right\rangle + \sum_{i=1}^{t} w_i \left\langle \theta_i, \mathbb{E}_i \widehat{l}_i - \widehat{l}_i \right\rangle.$$

The first term is bounded by

$$\begin{aligned} \sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \mathbb{E}_i \widehat{l}_i \right\rangle =& \sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \frac{\theta_i}{\theta_i + \gamma_i} l_i \right\rangle \\ =& \sum_{i=1}^{t} w_i \left\langle \theta_i, \frac{\gamma_i}{\theta_i + \gamma_i} l_i \right\rangle \le B \sum_{i=1}^{t} \gamma_i w_i. \end{aligned}$$

To bound the second term, notice

$$\left\langle \theta_i, \widehat{l}_i \right\rangle \le \sum_{b \in \mathcal{B}} \theta_i\left(b\right) \frac{\mathbb{I}\left\{b_t = b\right\}}{\theta_i(b) + \gamma_i} \le \sum_{b \in \mathcal{B}} \mathbb{I}\left\{b_i = b\right\} = 1,$$

thus $\{w_i \left\langle \theta_i, \mathbb{E}_i \widehat{l}_i - \widehat{l}_i \right\rangle\}_{i=1}^{t}$ is a bounded martingale difference sequence w.r.t. the filtration $\{\mathcal{F}_i\}_{i=1}^{t}$. By Azuma-Hoeffding,

$$\sum_{i=1}^{t} \left\langle \theta_i, \mathbb{E}_i \widehat{l}_i - \widehat{l}_i \right\rangle \le \sqrt{2\iota \sum_{i=1}^{t} w_i^2}.$$

□

Lemma 102. *For any $t \in [K]$, with probability $1-\delta$, for any $\theta^\star \in \Delta^B$, if γ_i is non-increasing in i,*

$$\sum_{i=1}^{t} w_i \left\langle \theta^\star, \widehat{l}_i - l_i \right\rangle \le \max_{i \le t} w_i \iota / \gamma_t.$$

Proof of Lemma 102. Define a basis $\{e_j\}_{j=1}^B$ of $\mathbb{R}^B$ by

$$e_j(b) = \begin{cases} 1 \text{ if } a = j \\ 0 \text{ otherwise} \end{cases}$$

Then for all the $j \in [B]$, we can apply Lemma 99 with $c_i = \gamma_t e_j$. Sincee $c_i(b) \leq \gamma_t \leq \gamma_i$, the condition in Lemma 99 is satisfied. As a result,

$$\sum_{i=1}^t w_i \left\langle e_j, \widehat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \iota / \gamma_t.$$

Since any $\theta^\star$ is a convex combination of $\{e_j\}_{j=1}^B$, by taking the union bound over $j \in [B]$, we have

$$\sum_{i=1}^t w_i \left\langle \theta^\star, \widehat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \iota / \gamma_t.$$

□

Finally we are ready to prove Theorem 98.

Proof of Theorem 98. Note the conditions in Lemma 100 and Lemma 102 are satisfied by assumptions. Recall the regret decomposition (C.8). By bounding (A) in Lemma 100, (B) in Lemma 101 and (C) in Lemma 102, with probability $1 - 3\delta$, we have that

$$\mathcal{R}(t) \leq \frac{w_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^t \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \iota + B \sum_{i=1}^t \gamma_i w_i + \sqrt{2\iota \sum_{i=1}^t w_i^2} + \max_{i \leq t} w_i \iota / \gamma_t.$$

□

C.7 Adversarial Bandit with Weighted Swap Regret

In this section, we adapt Follow-the-Regularized-Leader (FTRL) algorithm that achieves low weighted swap regret for the adversarial bandit problem. We follow a similar tech-

Algorithm 20 FTRL for Weighted Swap Regret (FTRL_swap)

```
**Initialize:** for any $b \in \mathcal{B}$, $\theta_1(b) \leftarrow 1/B$.
**for** episode $t = 1, \dots, K$ **do**
    Take action $b_t \sim \theta_t(\cdot)$, and observe loss $\tilde{l}_t(b_t)$.
    **for** each action $b \in \mathcal{B}$ **do**
        $\widehat{l}_t(\cdot|b) \leftarrow \theta_t(b)\tilde{l}_t(b_t)\mathbb{I}\{b_t = \cdot\}/(\theta_t(\cdot) + \gamma_t)$.
        $\tilde{\theta}_{t+1}(\cdot|b) \propto \exp[-(\eta_t/w_t) \cdot \sum_{i=1}^{t} w_i \widehat{l}_i(\cdot|b)]$
    Set $\theta_{t+1}$ such that $\theta_{t+1}(\cdot) = \sum_a \theta_{t+1}(b)\tilde{\theta}_{t+1}(\cdot|b)$.
```

nique presented in Blum and Mansour [2007] which adapts external regret algorithms to swap regret algorithms for the unweighted case.

We present the FTRL_swap algorithm in Algorithm 20. Different from FTRL (Algorithm 19), FTRL_swap maintains an additional $B \times B$ matrix $\tilde{\theta}_t(\cdot|\cdot)$, and uses its eigenvector when taking actions. The matrix will be updated similarly to FTRL, with a subtle difference that the loss estimator here $\widehat{\ell}_t(\cdot|b)$ is $\theta_t(b)$ times the loss estimator $\widehat{\ell}_t(\cdot)$ in the FTRL algorithm (Line 4 in Algorithm 19).

In Corollary 103, we prove that FTRL_swap satisfies the Assumption 2 with good swap regret bounds. Recall that B is the number of actions, and our normalization condition requires loss $\tilde{l}_t \in [0,1]^B$ for any t.

Corollary 103. *By choosing hyperparameter $w_t = \alpha_t \left(\prod_{i=2}^{t}(1-\alpha_i)\right)^{-1}$ and $\eta_t = \gamma_t = \sqrt{\frac{H \log B}{t}}$, FTRL_swap (Algorithm 20) satisfies Assumption 2 with*

$$\xi_{sw}(B, t, \log(1/\delta)) = 10B\sqrt{H \log(B^2/\delta)/t}, \quad \Xi_{sw}(B, t, \log(1/\delta)) = 20B\sqrt{Ht \log(B^2/\delta)}$$

Again, we prove Corollary 103 by showing a more general weighted swap regret guarantee which works for any set of weights $\{w_i\}_{i=1}^{\infty}$ in addition to $\{\alpha_t^i\}_{i=1}^{t}$. A general weighted swap regret is defined as

$$\mathcal{R}_{\mathrm{swap}}(t) := \min_{\psi \in \Psi} \sum_{i=1}^{t} w_i[\langle \theta_i, l_i \rangle - \langle \psi \diamond \theta_i, l_i \rangle]. \tag{C.9}$$

Theorem 104. *For any $t \le K$, following Algorithm 20, if $\eta_i \le 2\gamma_i$ and η_i is non-*

increasing for all $i \le t$, let $\iota = \log(B^2/\delta)$, then with probability $1 - 3\delta$, we have

$$\begin{aligned}
&\mathcal{R}_{swap}(t) \\
\le &\frac{w_t B \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t} \eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota + B\sum_{i=1}^{t}\gamma_i w_i + B\sqrt{2\iota \sum_{i=1}^{t} w_i^2} + \frac{B\iota}{\gamma_t}\max_{i\le t} w_i
\end{aligned}$$

We postpone the proof of Theorem 104 to the end of this section. We show first how Theorem 104 directly implies Corollary 103.

Proof of Corollary 103. As shown in the proof of Corollary 97, the weights $\{w_t\}_{t=1}^{K}$ we choose satisfies a nice property: for any t we have

$$\frac{w_i}{w_j} = \frac{\alpha_t^i}{\alpha_t^j}.$$

Define $\tilde{\mathcal{R}}_{\mathrm{swap}}(t) := \max_{\psi\in\Psi} \sum_{i=1}^{t} \alpha_t^i [\langle \theta_i, l_i\rangle - \langle \psi \diamond \theta_i, l_i\rangle]$. Plugging our choice of $w_i = \alpha_t \left(\prod_{i=2}^{t} (1-\alpha_i)\right)^{-1}$ into Theorem 104, we have

$$\begin{aligned}
\tilde{\mathcal{R}}_{\mathrm{swap}}(t) \le &\frac{\alpha_t B \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t}\eta_i \alpha_t^i + \frac{1}{2}\alpha_t \iota \\
&+ B\sum_{i=1}^{t}\gamma_i \alpha_t^i + B\sqrt{2\iota\sum_{i=1}^{t}\left(\alpha_t^i\right)^2} + B\alpha_t\iota/\gamma_t.
\end{aligned}$$

By choosing $\eta_t = \gamma_t = \sqrt{\frac{H\log B}{t}}$ and using Lemma 89, we can further upper bound the swap regret by

$$\begin{aligned}
\tilde{\mathcal{R}}_{\mathrm{swap}}(t) \le &\frac{(H+1)\, B\log B}{H+t}\sqrt{\frac{t}{H\log B}} + \frac{3B}{2}\sqrt{H\log B}\sum_{i=1}^{t}\frac{\alpha_t^i}{\sqrt{t}} \\
&+ \frac{(H+1)\,\iota}{2\,(H+t)} + B\sqrt{2\iota\sum_{i=1}^{t}\left(\alpha_t^i\right)^2} + B\frac{(H+1)\,\iota}{(H+t)}\sqrt{\frac{t}{H\log B}} \\
\le &2B\sqrt{H\frac{\log B}{t}} + 3B\sqrt{\frac{H\log B}{t}} + \frac{H\iota}{t} + 2B\sqrt{\frac{H\iota}{t}} + 2B\sqrt{\frac{H\log B}{t}} \\
\le &9B\sqrt{H\iota/t} + H\iota/t.
\end{aligned}$$

To further simplify the above upper bound, consider two cases:

- If $H\iota/t \leq 1$, $\sqrt{H\iota/t} \geq H\iota/t$ and thus $\tilde{\mathcal{R}}_{\text{swap}}(t) \leq 10B\sqrt{H\iota/t}$.
- If $H\iota/t \geq 1$, $B\sqrt{H\iota/t} \geq 1 \geq \tilde{\mathcal{R}}_{\text{swap}}(t)$ where the last step is by the definition of $\tilde{\mathcal{R}}_{\text{swap}}(t)$. Therefore we have $\tilde{\mathcal{R}}_{\text{swap}}(t) \leq B\sqrt{H\iota/t}$.

Combine the above two cases, $\tilde{\mathcal{R}}_{\text{swap}}(t) \leq 10B\sqrt{H\iota/t}$.

Finally, we pick $\xi_{\text{sw}}(B, t, \log(1/\delta)) := 10B\sqrt{H\iota/t}$, which is non-decreasing in B. On the other hand, since $\sum_{t'=1}^{t} \xi_{\text{sw}}(B, t, \log(1/\delta)) \leq 20B\sqrt{Ht\iota}$, we choose

$$\Xi_{\text{sw}}(B, t, \log(1/\delta)) = 20B\sqrt{Ht\iota},$$

which is concave in t. □

To prove Theorem 104, we again first decompose the swap regret. We first note that by Line 7 of Algorithm 20, we have:

$$w_i\langle \theta_i, l_i\rangle = \sum_{b\in\mathcal{B}} w_i\langle \tilde{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot)\rangle.$$

On the other hand, by the definition of strategy modification Ψ, we have

$$\min_{\psi\in\Psi}\sum_{i=1}^{t} w_i\langle \psi\diamond\theta_i, l_i\rangle = \sum_{b\in\mathcal{B}}\min_{\theta^\star(\cdot|b)}\sum_{i=1}^{t} w_i\theta_i(b)\cdot\langle \theta^\star(\cdot|b), l_i(\cdot)\rangle.$$

Therefore, we have the following decomposition of the swap regret

$$\begin{aligned}\mathcal{R}_{\text{swap}}(t) :=& \min_{\psi\in\Psi}\sum_{i=1}^{t} w_i[\langle \theta_i, l_i\rangle - \langle \psi\diamond\theta_i, l_i\rangle] = \sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i[\langle \tilde{\theta}_i(\cdot|b) - \theta^\star(\cdot|b), \theta_i(b)l_i\rangle] \\ =& \underbrace{\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\left\langle \tilde{\theta}_i(\cdot|b) - \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b)\right\rangle}_{(A)} \qquad\qquad (C.10) \\ &+\underbrace{\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot) - \widehat{l}_i(\cdot|b)\right\rangle}_{(B)}\end{aligned}$$

$$+\underbrace{\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\left\langle \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b) - \theta_i(b) l_i(\cdot)\right\rangle}_{(C)} \tag{C.11}$$

For the remaining proof, we bound term $(A), (B), (C)$ separately in Lemma 105, Lemma 106, Lemma 107.

Lemma 105. *For any $t \in [K]$, suppose $\eta_i \le 2\gamma_i$ for all $i \le t$. The with probability $1-\delta$, for any $\theta^\star$,*

$$\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\left\langle \tilde{\theta}_i(\cdot|b) - \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b)\right\rangle \le \frac{w_t B \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t}\eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota.$$

Proof of Lemma 105. Similar to Lemma 100, we have,

$$\begin{aligned}
&\sum_{i=1}^{t} w_i\left\langle \tilde{\theta}_i(\cdot|b) - \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b)\right\rangle \\
\le&\frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\eta_i w_i\left\langle \tilde{\theta}_i(\cdot|b), \widehat{l}_i^2(\cdot|b)\right\rangle \\
=&\frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b'\in\mathcal{B}}\eta_i w_i \tilde{\theta}_i(b'|b)\frac{\theta_i^2(b)\tilde{l}_i^2(b_i)\mathbb{I}\{b_i = b'\}}{(\theta_i(b') + \gamma_i)^2} \\
\le&\frac{w_t \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b'\in\mathcal{B}}\eta_i w_i \frac{\tilde{\theta}_i(b'|b)\theta_i(b)}{\theta_i(b')}\frac{\widehat{l}_i(b'|b)}{\theta_i(b)}
\end{aligned}$$

Summing over b and using the fact that $\sum_{b\in\mathcal{B}}\tilde{\theta}_i(b'|b)\theta_i(b) = \theta_i(b')$,

$$\begin{aligned}
\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\left\langle \tilde{\theta}_i(\cdot|b) - \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b)\right\rangle &\le \frac{w_t B \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b'\in\mathcal{B}}\eta_i w_i \frac{\widehat{l}_i(b'|b)}{\theta_i(b)} \\
&\overset{(i)}{\le} \frac{w_t B \log B}{\eta_t} + \frac{1}{2}\sum_{i=1}^{t}\sum_{b'\in\mathcal{B}}\eta_i w_i l_i(b') + \frac{1}{2}\max_{i\le t} w_i \iota \\
&\le \frac{w_t B \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^{t}\eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota
\end{aligned}$$

where (i) is by using Lemma 99 with $c_i(b) = \eta_i$. Notice the quantity $\frac{\widehat{l}_i(b'|b)}{\theta_i(b)}$ actually doesn't depend on b, so it is well-defined even after we take the summation with respect to b. The any-time guarantee is justified by taking union bound. □

Lemma 106. *For any $t \in [K]$, with probability $1-\delta$,*

$$\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b) l_i(\cdot) - \widehat{l}_i(\cdot|b) \right\rangle \le B\sum_{i=1}^{t}\gamma_i w_i + B\sqrt{2\iota \sum_{i=1}^{t} w_i^2}.$$

Proof of Lemma 106. We further decompose it into

$$\begin{aligned}&\sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b) l_i(\cdot) - \widehat{l}_i(\cdot|b) \right\rangle \\ =&\sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b) l_i(\cdot) - \mathbb{E}_i\widehat{l}_i(\cdot|b) \right\rangle + \sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \mathbb{E}_i\widehat{l}_i(\cdot|b) - \widehat{l}_i(\cdot|b) \right\rangle.\end{aligned}$$

The first term is bounded by

$$\begin{aligned}\sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b) l_i(\cdot) - \mathbb{E}_i\widehat{l}_i(\cdot|b) \right\rangle =&\sum_{i=1}^{t} w_i\theta_i(b) \left\langle \tilde{\theta}_i(\cdot|b), (1-\frac{\theta_i(\cdot)}{\theta_i(\cdot)+\gamma_i}) l_i(\cdot) \right\rangle \\ =&\sum_{i=1}^{t} w_i\theta_i(b) \left\langle \tilde{\theta}_i(\cdot|b), \frac{\gamma_i}{\theta_i(\cdot)+\gamma_i} l_i \right\rangle.\end{aligned}$$

So by taking the sum with respect to b, we have

$$\begin{aligned}\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i \left\langle \tilde{\theta}_i(\cdot|b), \theta_i(b) l_i(\cdot) - \mathbb{E}_i\widehat{l}_i(\cdot|b) \right\rangle \le&\sum_{b\in\mathcal{B}}\sum_{i=1}^{t} w_i\theta_i(b) \left\langle \tilde{\theta}_i(\cdot|b), \frac{\gamma_i}{\theta_i(\cdot)+\gamma_i} l_i \right\rangle \\ \le&\sum_{b'\in\mathcal{B}}\sum_{i=1}^{t} w_i\gamma_i l_i(b') \\ \le&B\sum_{i=1}^{t}\gamma_i w_i.\end{aligned}$$

To bound the second term, notice $\tilde{\theta}_i(b'|b)\theta_i(b) \le \theta_i(b')$ for any $b, b' \in \mathcal{B}$,

$$\left\langle \tilde{\theta}_i(\cdot|b), \widehat{l}_i(\cdot|b) \right\rangle \le \sum_{b'\in\mathcal{B}} \tilde{\theta}_i(b'|b)\theta_i(b)\frac{\mathbb{I}\{b_t = b'\}}{\theta_i(b')+\gamma_i} \le \sum_{b'\in\mathcal{B}} \mathbb{I}\{b_i = b'\} = 1,$$

thus $\{w_i \left\langle \tilde{\theta}_i(\cdot|b), \mathbb{E}_i\widehat{l}_i(\cdot|b) - \widehat{l}_i(\cdot|b) \right\rangle\}_{i=1}^t$ is a bounded martingale difference sequence w.r.t. the filtration $\{\mathcal{F}_i\}_{i=1}^t$. By Azuma-Hoeffding,

$$\sum_{i=1}^t w_i \left\langle \tilde{\theta}_i(\cdot|b), \mathbb{E}_i\widehat{l}_i(\cdot|b) - \widehat{l}_i(\cdot|b) \right\rangle \leq \sqrt{2\iota \sum_{i=1}^t w_i^2}.$$

The proof is completed by taking the summation with respect to b and a union bound. □

Lemma 107. *For any $t \in [K]$, suppose γ_i is non-increasing in i, then with probability $1-\delta$, and any $\theta^\star$,*

$$\sum_{b\in\mathcal{B}} \sum_{i=1}^t w_i \left\langle \theta^\star(\cdot|b), \widehat{l}_i(\cdot|b) - \theta_i(b) l_i(\cdot) \right\rangle \leq B \max_{i\leq t} w_i\iota/\gamma_t.$$

Proof of Lemma 107. The proof follows from Lemma 102 and taking the summation with respect to b. □

Finally, we are ready to prove Theorem 104.

Proof of Theorem 104. Recall the decomposition of swap regret (C.11). We bound (A) in Lemma 105, (B) in Lemma 106 and (C) in Lemma 107. Putting everything together, we have

$$\begin{aligned}&\mathcal{R}_{\text{swap}}(t)\\ \leq&\frac{w_t B \log B}{\eta_t} + \frac{B}{2}\sum_{i=1}^t \eta_i w_i + \frac{1}{2}\max_{i\leq t} w_i\iota + B\sum_{i=1}^t \gamma_i w_i + B\sqrt{2\iota\sum_{i=1}^t w_i^2} + \frac{B\max_{i\leq t} w_i\iota}{\gamma_t}.\end{aligned}$$

□

C.8 Proof for the V-OL algorithm

Throughout this section, let $\iota = \log(^{HSAK}/_{p})$.

C.8.1 Upper confidence bound on the minimax value function

Lemma 108 (V-learning lemma)**.** *In Algorithm 8, let $t = N_h^k(s)$ and suppose state $s \in \mathcal{S}_h$ was previously visited at episodes $k^1, \ldots, k^t < k$ at the hth step. For any $p \in (0,1)$, let $\iota = \log(HSAK/p)$. Choose $\eta_t = \sqrt{GH\log A/At}$. Then with probability at least $1-p$, for any $t \in [K]$, $h \in [H]$ and $s \in \mathcal{S}_h$, there exists a constant c such that*

$$\max_{\mu\in\Delta_{\mathcal{A}}} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu,\nu_h^{k^i}} \left[r_h + \mathbb{P}_h V_{h+1}^{k^i}\right](s) - \sum_{i=1}^{t} \alpha_t^i \left(r_h(s, a_h^{k^i}, b_h^{k^i}) + V_{h+1}^{k^i}(s_{h+1}^{k^i})\right) \le c\sqrt{\frac{GH^3A\iota}{t}}. \tag{C.12}$$

Proof. By the Azuma-Hoeffding inequality and Lemma 89,

$$\sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{h+1}^{k^i}\right)(s) - \sum_{i=1}^{t} \alpha_t^i \left[r_h\left(s, a_h^{k^i}, b_h^{k^i}\right) + V_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right)\right] \le 2\sqrt{\frac{GH^3\iota}{t}}.$$

So we only need to bound

$$R_t^* := \max_{\mu\in\Delta_{\mathcal{A}}} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu\times\nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{h+1}^{k^i}\right)(s) - \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i}\times\nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{h+1}^{k^i}\right)(s). \tag{C.13}$$

By taking $w_i = \alpha_t^i$ in [Bai et al., 2020, Lemma 17],

$$\begin{aligned}
R_t^* \le& \frac{3H\alpha_t^t \log A}{\eta_t} + \frac{3A}{2}\sum\nolimits_{i=1}^{t} \eta_i \alpha_t^i + \sqrt{2\iota \sum\nolimits_{i=1}^{t} \left(\alpha_t^i\right)^2} \\
\overset{(i)}{\le}& 3H\alpha_t^t \sqrt{\tfrac{At\log A}{GH}} + \frac{3}{2}\sqrt{GHA\log A}\sum\nolimits_{i=1}^{t} \frac{\alpha_t^i}{\sqrt{i}} + \sqrt{2\iota \sum\nolimits_{i=1}^{t} \left(\alpha_t^i\right)^2} \\
\overset{(ii)}{\le}& 3H\tfrac{GH+1}{GH+t}\sqrt{\tfrac{At\log A}{GH}} + 3\sqrt{\tfrac{GHA\log A}{t}} + 2\sqrt{\tfrac{GH\iota}{t}} \\
\le& c\sqrt{\tfrac{GHA\iota}{t}}
\end{aligned}$$

for some constant c, where (i) is by setting $\eta_t = \sqrt{\frac{GH\log A}{At}}$ and (ii) is by Lemma 89. Taking union bound w.r.t. all $(t, s, h) \in [K] \times \mathcal{S} \times [H]$ concludes the proof.

We comment that the quantity R_t^* is actually H times the LHS in the inequality of [Bai et al., 2020, Lemma 17]. See Appendix F and Algorithm 9 in Bai et al. [2020]

for a detailed reduction from MG to adversarial bandit problem. Furthermore, in [Bai et al., 2020] there are actually two parameters η_t and γ_t. Here we just take $\gamma_t = \eta_t$ for simplicity. Finally, the proof of [Bai et al., 2020, Lemma 17] requires that $\eta_i \leq 2\gamma_i$ for all $i \leq t$ [Bai et al., 2020, Lemma 19] and that γ_t is nondecreasing in t [Bai et al., 2020, Lemma 21], which are both satisfied by our specification of η_t. □

Lemma 109 (Upper confidence bound)**.** *In Algorithm 8, for any $p \in (0,1)$, let $\iota = \log({}^{HSAK}/_{p})$ and choose $\beta_t = c\sqrt{GH^3A\iota/t}$ for some large constant c. Then with probability at least $1-p$, $V_h^*(s) \leq V_h^k(s)$ for all $k \in [K]$, $h \in [H]$ and $s \in \mathcal{S}_h$.*

Proof. The proof is similar to that of [Bai et al., 2020, Lemma 15], except that we need to deal with an extra parameter G here.

Let $k_h^i(s)$ denote the index of the episode where $s \in \mathcal{S}_h$ is observed at step h for the ith time. Where there is no ambiguity, we use k^i as a shorthand for $k_h^i(s)$. Let s_h^k be the state actually observed in the algorithm at step h in episode k. For our choice of β_i, we have $\sum_{i=1}^t \alpha_t^i \beta_i = \Theta(GH^2\sqrt{{}^{A\iota}/_{t}})$ by Lemma 89.

Recall that

$$\begin{aligned} V_h^k(s) &:= \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left(r_h(s, a_h^{k^i}, b_h^{k^i}) + V_{h+1}^{k^i}(s_{h+1}^{k^i}) + \beta_i \right), \\ V_h^*(s) &:= \mathbb{D}_{\mu_h^*, \nu_h^*}[r_h + \mathbb{P}_h V_{h+1}^*](s). \end{aligned}$$

For $h = H+1$ the UCB vacuously holds. To apply backward induction, assume that $V_{h+1}^* \leq V_{h+1}^k$ holds entrywise. Then by definition, for any $s \in \mathcal{S}_h$,

$$\begin{aligned} V_h^*(s) &= \max_{\mu \in \Delta_{\mathcal{A}_h}} \min_{\nu \in \Delta_{\mathcal{B}_h}} \mathbb{D}_{\mu,\nu}[r_h + \mathbb{P}_h V_{h+1}^*](s) \\ &\overset{(i)}{=} \max_{\mu \in \Delta_{\mathcal{A}_h}} \sum\nolimits_{i=1}^t \alpha_t^i \min_{\nu \in \Delta_{\mathcal{B}_h}} \mathbb{D}_{\mu,\nu}[r_h + \mathbb{P}_h V_{h+1}^*](s) \\ &\leq \max_{\mu \in \Delta_{\mathcal{A}_h}} \sum\nolimits_{i=1}^t \alpha_t^i \mathbb{D}_{\mu,\nu_h^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^*](s) \\ &\overset{(ii)}{\leq} \max_{\mu \in \Delta_{\mathcal{A}_h}} \sum\nolimits_{i=1}^t \alpha_t^i \mathbb{D}_{\mu,\nu_h^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s) \overset{(iii)}{\leq} V_h^k(s), \end{aligned}$$

where (i) follows from $\sum_{i=1}^t \alpha_t^i = 1$, in (ii) we apply the induction assumption, and

(iii) holds with probability at least $1-p$ by the V-learning lemma (Lemma 108) and that $\sum_{i=1}^t \alpha_t^i \beta_t = \Theta(\sqrt{GH^3 A\iota/t})$ because of our choice of β_t and Property 1 of $\{\alpha_t^i\}$ in Lemma 89. Inductively we have $V_h^*(s) \le V_h^k(s)$ for all $k \in [K]$, $h \in [H]$ and $s \in \mathcal{S}_h$. □

C.8.2 Proof of Theorem 27

Proof. In the proof below, we use '$\lesssim$' to denote '$\le$' hiding some constants. Recall that

$$V_h^{\mu^k,\nu^k}(s_h^k) = \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k).$$

Then define $\delta_h^k := (V_h^k - V_h^{\mu^k,\nu^k})(s_h^k)$. By definition,

$$\begin{aligned}
\delta_h^k &= \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left(r_h(s_h^k, a_h^{k^i}, b_h^{k^i}) + V_{h+1}^{k^i}(s_{h+1}^{k^i}) + \beta_i \right) - \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k) \\
&\overset{(i)}{=} \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left(r_h(s_h^k, a_h^{k^i}, b_h^{k^i}) + V_{h+1}^{k^i}(s_{h+1}^{k^i}) + \beta_i \right) - \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu^{k^i},\nu^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s_h^k) \\
&\qquad + \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu^{k^i},\nu^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s_h^k) - \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k) \\
&\overset{(ii)}{\lesssim} \alpha_t^0 H + \sqrt{\tfrac{GH^3 A\iota}{t}} + \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu^{k^i},\nu^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s_h^k) - \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k),
\end{aligned}$$

where in (i) we add and subtract the same term, and (ii) follows from the property of β_i that $\sum_{i=1}^t \alpha_t^i \beta_i = \Theta(\sqrt{GH^3 A\iota/t})$ and the fact that by the Azuma-Hoeffding inequality and Property 2 of Lemma 89,

$$\sum_{i=1}^t \alpha_t^i \left(r_h(s_h^k, a_h^{k^i}, b_h^{k^i}) + V_{h+1}^{k^i}(s_{h+1}^{k^i}) \right) - \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu^{k^i},\nu^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s_h^k) \lesssim \sqrt{\tfrac{GH^3\iota}{t}}.$$

By the same regrouping technique as that in [Jin et al., 2018],

$$\sum_{k=1}^K \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu^{k^i},\nu^{k^i}}[r_h + \mathbb{P}_h V_{h+1}^{k^i}](s_h^k) \le \sum_{k'=1}^K \mathbb{D}_{\mu^{k'},\nu^{k'}}[r_h + \mathbb{P}_h V_{h+1}^{k'}](s_h^k) \sum_{t=n_h^{k'}}^{\infty} \alpha_t^{n_h^{k'}}$$

$$\le (1+\tfrac{1}{GH}) \sum_{k=1}^{K} \mathbb{D}_{\mu^k,\nu^k}[r_h + \mathbb{P}_h V_{h+1}^k](s_h^k).$$

Substituting the above back into the bound on δ_h^k and taking sum over $k \in [K]$, we obtain

$$\begin{aligned}
&\sum_{k=1}^{K} \delta_h^k \\
&\lesssim \sum_{k=1}^{K} \left(\alpha_t^0 H + \sqrt{\tfrac{GH^3 A\iota}{t}} + (1+\tfrac{1}{GH}) \mathbb{D}_{\mu^k,\nu^k}[r_h + \mathbb{P}_h V_{h+1}^k](s_h^k) - \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k) \right) \\
&\overset{(i)}{=} \sum_{k=1}^{K} \left(\alpha_t^0 H + \sqrt{\tfrac{GH^3 A\iota}{t}} + (1+\tfrac{1}{GH})(\delta_{h+1}^k + \gamma_h^k) + \tfrac{1}{GH} \mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k) \right) \\
&\overset{(ii)}{\le} \sum_{k=1}^{K} \left(\alpha_t^0 H + \sqrt{\tfrac{GH^3 A\iota}{t}} + (1+\tfrac{1}{GH})(\delta_{h+1}^k + \gamma_h^k) + \tfrac{1}{G} \right),
\end{aligned}$$

where in (i) we define the martingale difference term $\gamma_h^k := \mathbb{D}_{\mu_h^k,\nu_h^k}[\mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\mu^k,\nu^k})](s_h^k) - (V_{h+1}^k - V_{h+1}^{\mu^k,\nu^k})(s_{h+1}^k)$ and (ii) follows from that

$$\mathbb{D}_{\mu_h^k,\nu_h^k}[r_h + \mathbb{P}_h V_{h+1}^{\mu^k,\nu^k}](s_h^k) \le H.$$

Recursively,

$$\sum_{k=1}^{K} \delta_1^k \lesssim (1+\tfrac{1}{GH})^H \sum_{k=1}^{K} \sum_{h=1}^{H} \left(\alpha_t^0 H + \sqrt{\tfrac{GH^3 A\iota}{t}} + (1+\tfrac{1}{GH}) \gamma_h^k + \tfrac{1}{G} \right).$$

Now we bound each term in $\sum_{k=1}^{K} \delta_1^k$ separately by standard techniques in [Jin et al., 2018, Xie et al., 2020]:

$$\begin{aligned}
&\sum_{k=1}^{K} \alpha_{n_h^k}^0 H \le \sum_{k=1}^{K} H \cdot \mathbf{I}(n_h^k = 0) \le HS, \\
&\sum_{k=1}^{K} \sqrt{\tfrac{GH^3 A\iota}{n_h^k}} = GH^2\sqrt{A\iota} \sum_{k=1}^{K} \sqrt{\tfrac{1}{n_h^k}} \le \sqrt{GH^3 A\iota} \sum_{s \in \mathcal{S}_h} \sum_{n=1}^{n_h^K(s)} \sqrt{\tfrac{1}{n}} \lesssim \sqrt{GH^3 SAK\iota}), \\
&\sum_{k=1}^{K} \sum_{h=1}^{H} \gamma_h^k \lesssim \sqrt{H^3 K\iota},
\end{aligned}$$

where the second line follows from a pigeonhole argument and the third line follows from the Azuma-Hoeffding inequality. Combining the above bounds, we obtain

$$\mathrm{Regret}(K) \leq \sum_{k=1}^{K} \delta_1^k \lesssim H^2 S + \sqrt{GH^5 SAK\iota} + G^{-1} KH.$$

If $K \geq H^3 SA$ then we take we take $G = \frac{1}{H}(\frac{K}{SA})^{1/3}$; otherwise we take $G = K^{\frac{1}{3}}$. Then the following regret bounds holds:

$$\mathrm{Regret}(K) = \begin{cases} \tilde{\mathcal{O}}\big(H^2 S^{\frac{1}{3}} A^{\frac{1}{3}} K^{\frac{2}{3}} + H^2 S\big), & \text{if } K \geq H^3 SA, \\ \tilde{\mathcal{O}}\big(\sqrt{H^5 SA} K^{\frac{2}{3}} + H^2 S\big), & \text{otherwise.} \end{cases}$$

□

Appendix D

Proofs for Chapter 5

D.1 Properties of the game

D.1.1 Properties for Section 5.1

For any opponent (min-player) policy $\nu \in \Pi_{\min}$, define

$$p^{\nu}_{1:h}(x_h) := \sum_{s_h \in x_h} p_{1:h}(s_h)\nu_{1:h-1}(y(s_{h-1}), b_{h-1}) \quad \text{for all } h \in [H],\ x_h \in \mathcal{X}_h.$$

Intuitively, $p^{\nu}_{1:h}(x_h)$ measures the environment and the opponent's contribution in the reaching probability of x_h.

Lemma 110 (Properties of $p^{\nu}_{1:h}(x_h)$)**.** *The following holds for any $\nu \in \Pi_{\min}$:*

1. *For any policy $\mu \in \Pi_{\max}$, we have*

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:h}(x_h, a_h) p^{\nu}_{1:h}(x_h) = 1.$$

2. *$0 \le p^{\nu}_{1:h}(x_h) \le 1$ for all h, x_h.*

Proof. For (a), notice that

$$\mu_{1:h}(x_h, a_h) p^{\nu}_{1:h}(x_h) = \sum_{s_h \in x_h} p_{1:h}(s_h) \cdot \mu_{1:h}(x_h, a_h) \cdot \nu_{1:h-1}(y(s_{h-1}), b_{h-1})$$

$$= \sum_{s_h \in x_h} \mathbb{P}^{\mu,\nu}(\text{visit } (s_h, a_h)) = \mathbb{P}^{\mu,\nu}(\text{visit } (x_h, a_h)).$$

Summing over all $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$, the right hand side sums to one, thereby showing (a).

For (b), fix any $x_h \in \mathcal{X}_h$. Clearly $p^{\nu}_{1:h}(x_h) \geq 0$. Choose any $a_h \in \mathcal{A}$, and choose policy $\mu^{x_h,a_h} \in \Pi_{\max}$ such that $\mu^{x_h,a_h}_{1:h}(x_h, a_h) = 1$ (such μ^{x_h,a_h} exists, for example, by deterministically taking all actions prescribed in infoset x_h at all ancestors of x_h). For this μ^{x_h,a_h}, using (a), we have

$$p^{\nu}_{1:h}(x_h) = \mu^{x_h,a_h}_{1:h}(x_h, a_h) \cdot p^{\nu}_{1:h}(x_h) \leq \sum_{(x'_h, a'_h) \in \mathcal{X}_h \times \mathcal{A}} \mu^{x_h,a_h}_{1:h}(x'_h, a'_h) \cdot p^{\nu}_{1:h}(x'_h) = 1.$$

This shows part (b). □

Corollary 111. *For any policy $\mu \in \Pi_{\max}$ and $h \in [H]$, we have*

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:h}(x_h, a_h) \ell^t_h(x_h, a_h) \leq 1.$$

Proof. Notice by definition

$$\ell^t_h(x_h, a_h) = \sum_{s_h \in x_h, b_h \in \mathcal{B}_h} p_{1:h}(s_h) \nu^t_{1:h}(y(s_h), b_h)(1 - r_h(s_h, a_h, b_h)) \leq p^{\nu}_{1:h}(x_h),$$

and the result is implied by Lemma 110 (b). □

Lemma 112. *For any $h \in [H]$, the counterfactual loss function L^t_h defined in* (6.11) *satisfies the bound*

1. *For any policy $\mu \in \Pi_{\max}$, we have*

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:h}(x_h, a_h) L^t_h(x_h, a_h) \leq H - h + 1.$$

2. For any (h, x_h, a_h), we have

$$0 \le L_h^t(x_h, a_h) \le p_{1:h}^{\nu^t}(x_h) \cdot (H - h + 1).$$

Proof. Part (a) follows from the fact that

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:h}(x_h, a_h) L_h^t(x_h, a_h) = \mathbb{E}_{\mu, \nu^t}\left[\sum_{h'=h}^{H} r_{h'}\right] \le H - h + 1,$$

where the first equality follows from the definition of the loss functions ℓ_h and L_h in (5.5), (6.11).

For part (b), the nonnegativity follows clearly by definition. For the upper bound, take any policy $\mu^{x_h, a_h} \in \Pi_{\max}$ such that $\mu_{1:h}^{x_h, a_h}(x_h, a_h) = 1$. We then have

$$\begin{aligned}
L_h^t(x_h, a_h) &= \mu_{1:h}^{x_h, a_h}(x_h, a_h) L_h^t(x_h, a_h) = \mathbb{E}_{\mu^{x_h, a_h}, \nu^t}\left[\mathbf{1}\left\{\text{visit } x_h, a_h\right\} \cdot \sum_{h'=h}^{H} r_{h'}\right] \\
&= \mathbb{P}_{\mu^{x_h, a_h}, \nu^t}(\text{visit } x_h, a_h) \cdot \mathbb{E}_{\mu^{x_h, a_h}, \nu^t}\left[\sum_{h'=h}^{H} r_{h'} \middle| \text{visit } x_h, a_h\right] \\
&\le \mu_{1:h}^{x_h, a_h}(x_h, a_h) p_{1:h}^{\nu^t}(x_h) \cdot (H - h + 1) = p_{1:h}^{\nu^t}(x_h) \cdot (H - h + 1).
\end{aligned}$$

□

D.1.2 Properties for Section 5.4

For any $h < h'$ and $x_h \in \mathcal{X}_h$, we let $\mathcal{C}_{h'}(x_h, a_h) \equiv \{x \in \mathcal{X}_{h'} : x \succ (x_h, a_h)\}$ and $\mathcal{C}_{h'}(x_h) \equiv \{x \in \mathcal{X}_{h'} : x \succeq x_h\} = \cup_{a_h \in \mathcal{A}} \mathcal{C}_{h'}(x_h, a_h)$ denote the infosets within the h'-th step that are reachable from (i.e. children of) x_h or (x_h, a_h), respectively. For shorthand, let $\mathcal{C}(x_h, a_h) := \mathcal{C}_{h+1}(x_h, a_h)$ and $\mathcal{C}(x_h) := \mathcal{C}_{h+1}(x_h)$ denote the set of immediate children.

We define $X_{\succeq x_h}$ for any $x_h \in \mathcal{X}_h$ as

$$X_{\succeq x_h} := \sum_{h'=h}^{H} |\mathcal{C}_{h'}(x_h)|. \tag{D.1}$$

It can be interpreted as the number of infosets in the subtree rooted at x_h.

Lemma 113. *The L^1 norm of a sequence form is upper bounded by $\|\Pi\|_1 \le X$.*

Proof. We can prove $\|\Pi^{x_h}\| \le X_{\succeq x_h}$ for all $h \in [0, H]$ and $x_h \in \mathcal{X}_h$ by backward induction over $h = H, \cdots, 1, 0$. When $h = H$, for each infoset x_h, the sequence form is just a probability distribution, which sums up to $\|\Pi^{x_h}\|_1 = 1 = |X_{\succeq x_h}|$. If the claim holds for $h+1$, consider an infoset x_h in the h-th level. By induction hypothesis we have

$$\|\Pi^{x_h}\|_1 = \max_{a_h} \sum_{x_{h+1} \succeq (x_h, a_h)} \|\Pi^{x_{h+1}}\|_1 \le \max_{a_h} \sum_{x_{h+1} \succeq (x_h, a_h)} |X_{\succeq x_{h+1}}| \le |X_{\succeq x_h}|.$$

So the equation above holds for any x_h. Setting $x_h = \emptyset$ gives $\|\Pi\|_1 \le X$ which completes the proof. □

Lemma 114. *We have $|\Phi_0^{\mathsf{EFCE}}| \le X A^{\|\Pi\|_1 + 1}$.*

Proof. By Proposition 5.1 of Farina et al. [2022b], $\mathcal{V} \le A^{\|\Pi\|_1}$. Since there are at most XA different infoset-action pair to be trigger, we have $|\Phi_0^{\mathsf{EFCE}}| \le X A^{\|\Pi\|_1 + 1}$. □

D.2 Bounds for regret minimizers

Here we collect regret bounds for various regret minimization algorithms on the probability simplex. For any algorithm that plays policy p_t in the t-th round and observes loss vector $\{\ell_t(a)\}_{a \in [A]} \in \mathbb{R}^A_{\ge 0}$, define its regret as

$$\mathrm{Regret}(T) := \max_{p^\star \in \Delta([A])} \sum_{t=1}^{T} \left\langle p_t, \widetilde{\ell}_t \right\rangle - \left\langle p^\star, \widetilde{\ell}_t \right\rangle.$$

D.2.1 Hedge

The following regret bound for Hedge is standard, see, e.g. [Lattimore and Szepesvári, 2020, Proposition 28.7].

Algorithm 21 Regret Minimization with Hedge (Hedge)

```
Require: Learning rate $\eta > 0$.
Initialize $p_1(a) \leftarrow 1/A$ for all $a \in [A]$.
for iteration $t = 1, \dots, T$ do
    Receive loss vector $\left\{\widetilde{\ell}_t(a)\right\}_{a \in [A]}$.
    Update action distribution via mirror descent:
        $p_{t+1}(a) \propto_a p_t(a) \exp\left(-\eta \widetilde{\ell}_t(a)\right).$
```

Lemma 115 (Regret bound for Hedge)**.** *Algorithm 21 with learning rate $\eta > 0$ achieves regret bound*

$$\mathrm{Regret}(T) \le \frac{\log A}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a \in [A]} p_t(a) \widetilde{\ell}_t(a)^2.$$

D.2.2 Regret Matching

Algorithm 22 Regret Minimization with Regret Matching (RegretMatching)

```
Initialize $p_1(a) \leftarrow 1/A$ and $R_0(a) \leftarrow 0$ for all $a \in [A]$.
for iteration $t = 1, \dots, T$ do
    Receive loss vector $\left\{\widetilde{\ell}_t(a)\right\}_{a \in [A]}$.
    Update instantaneous regret and cumulative regret for all $a \in [A]$:
        $r_t(a) \leftarrow \left\langle p_t, \widetilde{\ell}_t \right\rangle - \widetilde{\ell}_t(a) \quad \text{and} \quad R_t(a) \leftarrow R_{t-1}(a) + r_t(a).$
    Compute action distribution by regret matching:
        $p_{t+1}(a) \leftarrow \frac{[R_t(a)]_+}{\sum_{a' \in [A]} [R_t(a')]_+} = \frac{\left[\sum_{t=1}^{T} \left\langle p_t, \tilde{\ell}_t \right\rangle - \widetilde{\ell}_t(a)\right]_+}{\sum_{a' \in [A]} \left[\sum_{t=1}^{T} \left\langle p_t, \tilde{\ell}_t \right\rangle - \widetilde{\ell}_t(a')\right]_+}.$
    In the edge case where $[R_t(a)]_+ = 0$ for all $a \in [A]$, set $p_{t+1}(a) \leftarrow 1/A$ to be
    the uniform distribution.
```

The following regret bound for Regret Matching is standard, see, e.g. [Cesa-Bianchi and Lugosi, 2006, Brown and Sandholm, 2014]. For completeness, here we provide a proof along with an alternative form of bound useful for our purpose (Remark 117). Note that here η is not the learning rate but rather an arbitrary positive value (i.e. the right-hand side is an upper bound on the regret for any $\eta > 0$). Algorithm 22 itself does not require any learning rate.

Lemma 116 (Regret bound for Regret Matching)**.** *Algorithm 22 achieves the following regret bound for* any $\eta > 0$*:*

$$\mathrm{Regret}(T) \le \Big[\sum_{t=1}^{T}\sum_{a\in[A]}\Big(\Big\langle p_t, \widetilde{\ell}_t\Big\rangle - \widetilde{\ell}_t(a)\Big)^2\Big]^{1/2} \le \frac{1}{\eta} + \frac{\eta}{4}\sum_{t=1}^{T}\sum_{a\in[A]}\Big(\Big\langle p_t, \widetilde{\ell}_t\Big\rangle - \widetilde{\ell}_t(a)\Big)^2.$$

Proof. By the fact that $(a+b)_+^2 \le a_+^2 + 2a_+b + b^2$, we have

$$[R_t(a)]_+^2 \le [R_{t-1}(a)]_+^2 + 2[R_{t-1}(a)]_+ r_t(a) + r_t(a)^2. \tag{D.2}$$

Then by the definition of $p_t(a)$ and $r_t(a)$, we have

$$\begin{aligned}\sum_{a\in[A]}[R_{t-1}(a)]_+ r_t(a) &= \sum_{a\in[A]}[R_{t-1}(a)]_+\Big(\sum_{a'\in[A]} p_t(a')\widetilde{\ell}_t(a') - \widetilde{\ell}_t(a)\Big)\\ &= \sum_{a\in[A]}[R_{t-1}(a)]_+\widetilde{\ell}_t(a) - \sum_{a\in[A]}[R_{t-1}(a)]_+\widetilde{\ell}_t(a) = 0.\end{aligned} \tag{D.3}$$

Then summing over a in Eq. (D.2) and using Eq. (D.3), we get

$$\begin{aligned}\sum_{a\in[A]}[R_T(a)]_+^2 \le& \sum_{a\in[A]}[R_{T-1}(a)]_+^2 + 2\sum_{a\in[A]}[R_{T-1}(a)]_+ r_T(a) + \sum_{a\in[A]} r_T(a)^2\\ =& \sum_{a\in[A]}[R_{T-1}(a)]_+^2 + \sum_{a\in[A]} r_T(a)^2 \le \sum_{t=1}^{T}\sum_{a\in[A]} r_t(a)^2.\end{aligned}$$

Using that $\max_a R_T(a) \le \max_a [R_T(a)]_+ \le (\sum_{a\in[A]}[R_T(a)]_+^2)^{1/2}$ gives the regret bound

$$\mathrm{Regret}(T) = \max_{a\in[A]} R_T(a) \le \left(\sum_{t=1}^{T}\sum_{a\in[A]} r_t(a)^2\right)^{1/2} = \left(\sum_{t=1}^{T}\sum_{a\in[A]}\Big(\Big\langle p_t, \widetilde{\ell}_t\Big\rangle - \widetilde{\ell}_t(a)\Big)^2\right)^{1/2}.$$

The claimed bound with η follows directly from the inequality $\sqrt{z} \le 1/\eta + \eta z/4$ for any $\eta > 0$, $z \geq 0$. □

Remark 117. *The quantity* $\sum_{a\in[A]}\left(\left\langle p_t, \widetilde{\ell}_t\right\rangle - \widetilde{\ell}_t(a)\right)^2$ *above can be upper bounded as*

$$\begin{aligned}
&\sum_{a\in[A]}\left(\left\langle p_t, \widetilde{\ell}_t\right\rangle - \widetilde{\ell}_t(a)\right)^2 \leq \sum_{a\in[A]}\left(\left\langle p_t, \widetilde{\ell}_t\right\rangle^2 + \widetilde{\ell}_t(a)^2\right) \\
&= A\left\langle p_t, \widetilde{\ell}_t\right\rangle^2 + \|\widetilde{\ell}_t\|_2^2 \leq A\sum_{a\in[A]}\left(p_t(a)\widetilde{\ell}_t(a)^2 + (1/A)\widetilde{\ell}_t(a)^2\right) \\
&= 2A\sum_{a\in[A]}\overline{p}_t(a)\widetilde{\ell}_t(a)^2,
\end{aligned}$$

where $\overline{p}_t(a) = [p_t(a) + (1/A)]/2$ *is a probability distribution over* $[A]$.

As a consequence, we get an upper bound on the regret of Regret Matching algorithm by

$$\mathrm{Regret}(T) \leq \frac{1}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{a\in[A]}(A\overline{p}_t(a))\widetilde{\ell}_t(a)^2.$$

Comparing to the bound of Hedge (Lemma 115), the above regret bound for Regret Matching has a similar form except for replacing $\log A$ *by* 1 *and replacing* p_t *by* $A\overline{p}_t$.

D.2.3 Φ-Hedge

The following lemma is standard and gives a Φ-regret bound of the Φ-Hedge algorithm.

Lemma 118 (Regret bound for Φ-Hedge)**.** *For strategy modification vertex set* Φ_0, *step size* η, *and total steps* T, *running Algorithm 9 gives*

$$\mathrm{Reg}^{\Phi}(T) \leq \frac{\log|\Phi_0|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0} p_\phi^t\big(\langle\phi\mu^t, \ell^t\rangle\big)^2.$$

Proof. We have

$$\mathrm{Reg}^{\Phi}(T) = \sup_{\phi\in\Phi}\sum_{t=1}^{T}\langle\mu^t - \phi\mu^t, \ell^t\rangle \overset{(i)}{=} \sup_{\phi\in\Phi}\sum_{t=1}^{T}\langle\phi^t\mu^t - \phi\mu^t, \ell^t\rangle$$

$$\overset{(ii)}{=} \sup_{p \in \Delta_{\Phi_0}} \sum_{t=1}^{T} \sum_{\phi \in \Phi_0} \big(p_\phi^t \langle \phi \mu^t, \ell^t \rangle - p_\phi \langle \phi \mu^t, \ell^t \rangle \big).$$

Above, (i) uses the fixed point equation $\phi^t \mu^t = \mu^t$ (Line 4), and (ii) uses the fact that $\Phi = \mathrm{conv}\{\Phi_0\}$. Note that the above expression is exactly the regret of $\{p^t\}_{t=1}^T$, where the loss vector in the t-th round is $\{\langle \phi \mu^t, \ell^t \rangle\}_{\phi \in \Phi_0}$. Further, the update rule of p^t (Line 6) coincides with Hedge algorithm. So by the standard regret bound for Hedge, see, e.g. (Lattimore and Szepesvári [2020], Proposition 28.7), we have

$$\begin{aligned} \mathrm{Reg}^{\Phi}(T) &= \sup_{p \in \Delta_{\Phi_0}} \sum_{t=1}^{T} \sum_{\phi \in \Phi_0} \big(p_\phi^t \langle \phi \mu^t, \ell^t \rangle - p_\phi \langle \phi \mu^t, \ell^t \rangle \big) \\ &\le \frac{\log |\Phi_0|}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{\phi \in \Phi_0} p_\phi \big(\langle \phi \mu^t, \ell^t \rangle \big)^2. \end{aligned}$$

This proves the lemma. □

The following Freedman's inequality can be found in [Agarwal et al., 2014, Lemma 9].

Lemma 119 (Freedman's inequality). *Suppose random variables $\{X_t\}_{t=1}^T$ is a martingale difference sequence, i.e. $X_t \in \mathcal{F}_t$ where $\{\mathcal{F}_t\}_{t \ge 1}$ is a filtration, and $\mathbb{E}[X_t | \mathcal{F}_{t-1}] = 0$. Suppose $X_t \le R$ almost surely for some (non-random) $R > 0$. Then for any $\lambda \in (0, 1/R]$, we have with probability at least $1 - \delta$ that*

$$\sum_{t=1}^{T} X_t \le \lambda \cdot \sum_{t=1}^{T} \mathbb{E}\big[X_t^2 | \mathcal{F}_{t-1}\big] + \frac{\log(1/\delta)}{\lambda}.$$

D.3 Equivalence to classical definitions of EFGs

We first formally define Extensive-Form Games (EFGs). We then show that solving EFGs with adversarial opponents can be reduced to solving Tree-Formed AMDP. See Section 5.1 for the classical definition of IIEFGs.

D.3.1 Reduction from classical definition of EFGs to TFAMDP

In this section, we show that solving EFGs with adversarial opponents can be reduced to solving TFAMDP. Formally, we prove the following proposition.

Proposition 120. *For any EFG (i.e. POMG with tree structure and perfect recall assumptions)* $(H, \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$ *with adversarial opponents' policies* $\{\nu^t\}_{t\geq 1}$*, there exists an adversarial MDP* $(\tilde{H}, \tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{\mathcal{T}})$ *with adversarial transition* $\{\tilde{p}^t = \{p_h^t\}_{h\in\{0\}\cup[H]}\}_{t\geq 1}$ *and reward* $\{\tilde{R}^t = \{\tilde{R}_h^t\}_{h\in[H]}\}_{t\geq 1}$*, so that for any policy sequences* $\{\mu^t\}_{t\geq 1}$*, their joint distributions over the learner's trajectory* $P(x_1, a_1, x_2, a_2, \ldots, x_H, a_H, r)$ *are exactly the same for all episodes* $t \geq 1$*.*

We remark that the joint distribution $P(x_1, a_1, x_2, a_2, \ldots, x_H, a_H, r)$ gives a complete description about what the first player can obtain from the dynamic systems in both models. The joint distributions being the same for two models means that information-theoretically, the learner has no way to distinguish the two models, thus proving their equivalence.

Proof of Proposition 120. In this section, we will use the notation in its original form $\mathcal{X}, \mathcal{A}, p, r$ to denote the quantity in EFGs while use their tilded form $\tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{p}, \tilde{R}$ to denote the corresponding quantity in tree-form AMDP. It is not hard to see that in order to prove Proposition 120 for all $t \geq 1$, it suffices to prove for a fixed t it is true.

Construction of AMDP we construct the corresponding AMDP using EFGs in the following way: we let $\tilde{H} = H$, $\tilde{\mathcal{X}} = \mathcal{X}$, $\tilde{\mathcal{A}} = \mathcal{A}$. Since EFG satisfies perfect recall assumption, which defines the immediate children function $\mathcal{C}$. We use the precisely same child function to define the tree structure $\tilde{\mathcal{T}}$ in AMDP. We define the adversarial transition according to the following equations:

$$\begin{aligned}\tilde{p}_1^t(x_1) :=& \sum_{s_1\in x_1} p_1(s_1),\\ \tilde{p}_h^t(x_{h+1}|x_h, a_h) :=& \frac{\sum_{s_{h+1}\in x_{h+1}} \sum_{b_{h+1}\in\mathcal{B}} p_{1:h+1}(s_{h+1})\nu_{1:h+1}^t(y_{h+1}(s_{h+1}), b_{h+1})}{\sum_{s_h\in x_h} \sum_{b_h\in\mathcal{B}} p_{1:h}(s_h)\nu_{1:h}^t(y_h(s_h), b_h)},\end{aligned}$$

where $y_{h+1}(s_{h+1})$ and $y_h(s_h)$ are the infoset of opponent at $(h+1)$-th and h-th steps given state s_{h+1} and s_h respectively. We also define the adversarial reward distribution $\tilde{R}^t_H(\cdot|x_H, a_H)$ such that it gives the following distribution over reward $r \in [0,1]$ for any fixed (x_H, a_H)

$$r = r(s_H, a_H, b_H) \text{ with probability } \frac{p_{1:H}(s_H)\nu^t_{1:H}(y_H(s_H), b_H)}{\sum_{s'_H \in x_H} \sum_{b'_H \in \mathcal{B}} p_{1:H}(s'_H)\nu^t_{1:H}(y_H(s'_H), b'_H)}.$$

And we set the adversarial reward $\tilde{R}^t_h(\cdot|x_h, a_h)$ to be zero (almost surely) for all $h \le H-1$ and all (x_h, a_h, t).

Proof of equivalence Denote $\tilde{P}^{\mu,t}$ as the probability of AMDP at episode t with policy μ; denote P^{μ,ν^t} as the probability of EFGs under policy μ and ν^t. It is very easy to check by induction over step h, that for any $h \in [H]$, and all policy μ simultaneously:

$$\tilde{P}^{\mu,t}(x_h, a_h) = P^{\mu,\nu^t}(x_h, a_h) = \sum_{s_h \in x_h} \sum_{b_h \in \mathcal{B}} \mu_{1:h}(x_h, a_h) p_{1:h}(s_h) \nu^t_{1:h}(y_h(s_h), b_h).$$

This proves that the joint distribution:

$$\tilde{P}^{\mu,t}(x_1, a_1, \ldots, x_H, a_H) = P^{\mu,\nu^t}(x_1, a_1, \ldots, x_H, a_H). \tag{D.4}$$

Finally, the construction of adversarial reward is such that its conditional distribution given (x_H, a_H) is exactly the same as the conditional distribution of the reward in the EFG:

$$\tilde{R}^t_H(r = r(s_H, a_H, b_H)|x_H, a_H) = P^{\mu,\nu^t}(r = r(s_H, a_H, b_H)|x_H, a_H),$$

which immediately gives that:

$$\tilde{P}^{\mu,t}(r_H|x_1, a_1, \ldots, x_H, a_H) = P^{\mu,\nu^t}(r|x_1, a_1, \ldots, x_H, a_H). \tag{D.5}$$

Combining (D.4) and (D.5), we finish the proof. □

D.4 Proof of Theorem 35

Both the regret and PAC lower bounds follow from a direct reduction to stochastic multi-armed bandits. For completeness, we first state the lower bound for stochastic bandits [Lattimore and Szepesvári, 2020, Exercise 15.4 & Exercise 33.1] as follows. Below, c is an absolute constant.

Proposition 121 (Lower bound for stochastic bandits)**.** *Let $K \geq 2$ denote the number of arms.*

1. *(Regret lower bound) Suppose $T \geq K$. For any bandit algorithm that plays policy $\mu^t \in \Delta([K])$ (either deterministic or random) in round $t \in [T]$, there exists some K-armed stochastic bandit problem with Bernoulli rewards with mean vector $r \in [0,1]^K$, on which the algorithm suffers from the following lower bound on the expected regret:*

$$\mathbb{E}\left[\max_{\mu^\dagger \in \Delta([K])} \sum_{t=1}^{T} \left\langle \mu^\dagger - \mu^t, r \right\rangle\right] \geq c \cdot \sqrt{KT}.$$

2. *(PAC lower bound) For any bandit algorithm that plays for t rounds and outputs some policy $\widehat{\mu} \in \Delta([K])$, there exists some K-armed stochastic bandit problem with Bernoulli rewards with some mean vector $r \in [0,1]^K$, on which policy $\widehat{\mu}$ is at least ε away from optimal:*

$$\mathbb{E}\left[\max_{\mu^\dagger \in \Delta([K])} \left\langle \mu^\dagger - \widehat{\mu}, r \right\rangle\right] \geq \varepsilon,$$

unless $T \geq cK/\varepsilon^2$.

We now construct a class of IIEFGs with $X_H = A^{H-1}$ (the minimal possible number of infosets), and show that any algorithm that solves this class of games will imply an algorithm for stochastic bandits with A^H arms with the same regret/PAC bounds, from which Theorem 35 follows.

Our construction is as follows: For any $A \geq 2$ and $H \geq 1$, we let $S_h = A^{h-1}$ for all $h \in [H]$ (in particular, $S_1 = 1$) and $B = 1$ (so that there is no opponent effectively).

By the tree structure, each state is thus uniquely determined by all past actions $s_h = (a_1, \dots, a_{h-1})$, and the transition is deterministic: $((a_1, \dots, a_{h-1}), a_h) \in \mathcal{S}_h \times \mathcal{A}$ transits to $(a_1, \dots, a_h) \in \mathcal{S}_{h+1}$ with probability one. Further, we let $x_h = x(s_h) = s_h$, so that there is no partial observability, and thus $\mathcal{X}_h = \mathcal{S}_h$ for all h. Only the H-th layer yields a Bernoulli reward with some mean $r_{a_{1:H}} := \mathbb{E}[r_H(a_{1:H-1}, a_H)] \in [0, 1]$, for all $a_{1:H} \in \mathcal{X}_H$. The reward is zero within all previous layers.

Under this model, the expected reward under any policy $\mu \in \Pi_{\max}$ can be succinctly written as

$$\langle \mu, r \rangle = \sum_{(x_H, a_H) \in \mathcal{X}_H \times \mathcal{A}} \mu_{1:H}(x_H, a_H)\mathbb{E}[r_H(x_h, a_H)] = \sum_{a_{1:H} \in \mathcal{A}^H} \mu_{1:H}(a_{1:H}) r_{a_{1:H}}.$$

This expression coincides with the expression for the expected reward of an A^H-armed stochastic bandit problem.

Now, for any algorithm Alg achieving regret $\mathfrak{R}^T$ on IIEFGs, we claim we can use it to design an algorithm for solving any A^H-armed stochastic bandit problem with Bernoulli rewards, and achieve the same regret. Indeed, given any A^H-armed bandit problem, we rename its arms as a sequence $a_{1:H} = (a_1, \dots, a_H) \in \mathcal{A}^H$. Now, we instantiate an instance of Alg on a simulated IIEFG with the above structure. Whenever Alg plays policy $\mu^t \in \Pi_{\max}$, we query an arm $a_{1:H}$ using policy $\mu^t_{1:H}(\cdot) \in \Delta(\mathcal{A}^H)$ in the bandit problem. Then, upon receiving the reward r^t from the bandit problem, we give the feedback that the game transitted to infoset $a_{1:H}$ and yielded reward r^t. By the above equivalence, the regret $\mathfrak{R}^T$ within this simulated game is exactly the same as the regret for the bandit problem.

Therefore, for $T \geq A^H$, we can apply Proposition 121(a) to show that for any such Alg, there exists one such IIEFG, on which

$$\mathbb{E}\big[\mathfrak{R}^T\big] \geq c \cdot \sqrt{A^H T} = c\sqrt{X_H A T} \geq c\sqrt{XAT},$$

where the last inequality follows from the fact that $X \leq X_H(1 + 1/A + 1/A^2 + \cdots) \leq X_H/(1 - 1/A) \leq 2X_H$ by perfect recall. This shows part (a).

Part (b) (PAC lower bound) follows similarly from Proposition 121(b). Using the same reduction, we can show for any algorithm that controls both players and outputs policy $(\widehat{\mu}, \widehat{\nu}) \in \Pi_{\max} \times \Pi_{\min}$, there exists one such game of the above form (where only the max player affects the game) where the algorithm suffers from the PAC lower bound

$$\mathbb{E}[\mathrm{NEGap}(\widehat{\mu}, \widehat{\nu})] = \mathbb{E}\left[\max_{\mu \in \Pi_{\max}} V^{\mu^{\dagger}, \widehat{\nu}} - V^{\widehat{\mu}, \widehat{\nu}}\right] \geq \varepsilon$$

unless $T \geq cXA/\varepsilon^2$. The symmetrical construction for the min player implies that there exists some game on which $\mathbb{E}[\mathrm{NEGap}(\widehat{\mu}, \widehat{\nu})] \geq \varepsilon$ unless $T \geq cYB/\varepsilon^2$.

Therefore, if $T < c(XA+YB)/(2\varepsilon^2)$, at least one of $T \geq cXA/\varepsilon^2$ and $T \geq cYB/\varepsilon^2$ has to be false, for which we obtain a game where the expected duality gap is at least ε. This shows part (b). □

Appendix E

Proofs for Chapter 6

E.1 Proofs of balanced exploration policy

Proof of Lemma 37. We have

$$
\begin{aligned}
&\sum_{x_h,a_h} \frac{\mu_{1:h}(x_h,a_h)}{\mu^{\star,h}_{1:h}(x_h,a_h)} \\
&= \sum_{x_{h-1},a_{h-1}} \sum_{(x_h,a_h)\in\mathcal{C}(x_{h-1},a_{h-1})\times\mathcal{A}} \frac{\mu_{1:(h-1)}(x_{h-1},a_{h-1})\cdot\mu_h(a_h|x_h)}{\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})\cdot(1/A)} \\
&\overset{(i)}{=} A\cdot\sum_{x_{h-1},a_{h-1}} \sum_{x_h\in\mathcal{C}(x_{h-1},a_{h-1})} \frac{\mu_{1:(h-1)}(x_{h-1},a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})} \\
&= A\cdot\sum_{x_{h-1},a_{h-1}} \frac{\mu_{1:(h-1)}(x_{h-1},a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})}\cdot|\mathcal{C}_h(x_{h-1},a_{h-1})| \\
&\overset{(ii)}{=} A\cdot\sum_{x_{h-2},a_{h-2}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}(x_{h-2},a_{h-2})\times\mathcal{A}} \\
&\qquad \frac{\mu_{1:(h-2)}(x_{h-2},a_{h-2})\mu_{h-1}(a_{h-1}|x_{h-1})}{\mu^{\star,h}_{1:(h-2)}(x_{h-2},a_{h-2})\cdot|\mathcal{C}_h(x_{h-1},a_{h-1})|/|\mathcal{C}_h(x_{h-1})|}\cdot|\mathcal{C}_h(x_{h-1},a_{h-1})| \\
&= A\cdot\sum_{x_{h-2},a_{h-2}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}(x_{h-2},a_{h-2})\times\mathcal{A}} \frac{\mu_{1:(h-2)}(x_{h-2},a_{h-2})\mu_{h-1}(a_{h-1}|x_{h-1})}{\mu^{\star,h}_{1:(h-2)}(x_{h-2},a_{h-2})}\cdot|\mathcal{C}_h(x_{h-1})| \\
&= A\cdot\sum_{x_{h-2},a_{h-2}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}(x_{h-2},a_{h-2})\times\mathcal{A}} \frac{\mu_{1:(h-2)}(x_{h-2},a_{h-2})\mu_{h-1}(a_{h-1}|x_{h-1})}{\mu^{\star,h}_{1:(h-2)}(x_{h-2},a_{h-2})}\cdot|\mathcal{C}_h(x_{h-1})| \\
&\overset{(iii)}{=} A\cdot\sum_{x_{h-2},a_{h-2}} \frac{\mu_{1:(h-2)}(x_{h-2},a_{h-2})}{\mu^{\star,h}_{1:(h-2)}(x_{h-2},a_{h-2})}\cdot|\mathcal{C}_h(x_{h-2},a_{h-2})|
\end{aligned}
$$

$$
\begin{aligned}
&= \dots \\
&= A \cdot \sum_{x_1, a_1} \frac{\mu_1(a_1|x_1)}{|\mathcal{C}_h(x_1, a_1)| / |\mathcal{C}_h(x_1)|} \cdot |\mathcal{C}_h(x_1, a_1)| \\
&= A \cdot \sum_{x_1, a_1} \mu_1(a_1|x_1) \cdot |\mathcal{C}_h(x_1)| \\
&= A \cdot \sum_{x_1} |\mathcal{C}_h(x_1)| = A \cdot |\mathcal{C}_h(\emptyset)| = X_h A.
\end{aligned}
$$

Above, (i) used the definition of $\mu_h^{\star,h}$ and the fact that $\sum_{a_h \in \mathcal{A}} \mu_h(a_h|x_h) = 1$ for any μ, x_h; (ii) used the definition of $\mu_{h-1}^{\star,h}$; (iii) used the fact that $\sum_{x_{h-1} \in \mathcal{C}(x_{h-2}, a_{h-2})} |\mathcal{C}_h(x_{h-1})| = |\mathcal{C}_h(x_{h-2}, a_{h-2})|$ which follows by the additivity of the number of descendants; and the rest followed by performing the same operations repeatedly. □

The following corollary is similar to the lower bound in [Farina et al., 2020b, Appendix A.3].

Corollary 122. *We have*

$$\mu_{1:h}^{\star,h}(x_h, a_h) \geq \frac{1}{X_h A}$$

for any $h \in [H]$ and $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$.

Proof. Choose some deterministic policy μ s.t. $\mu_{1:h}(x_h, a_h) = 1$ in Lemma 37 and noticing each term in the summation is non-negative,

$$\frac{\mu_{1:h}(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \leq X_h A.$$

□

E.2 Proofs for Balanced dilated KL

Proof of Lemma 40. We have

$$\max_{\mu^\dagger \in \Pi_{\max}} \mathrm{D}^{\mathrm{bal}}(\mu^\dagger \| \mu^{\mathrm{unif}}) = \max_{\mu^\dagger \in \Pi_{\max}} \sum_{h=1}^{H} \sum_{x_h, a_h} \frac{\mu_{1:h}^\dagger(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \log \frac{\mu_h^\dagger(a_h|x_h)}{\mu_h^{\mathrm{unif}}(a_h|x_h)}$$

$$
\begin{aligned}
&= \max_{\mu^\dagger \in \Pi_{\max}} \sum_{h=1}^{H} \sum_{x_h, a_h} \frac{\mu^\dagger_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \Big(\log \mu^\dagger_h(a_h|x_h) + \log A \Big) \\
&\overset{(i)}{\le} \log A \sum_{h=1}^{H} \max_{\mu^\dagger \in \Pi_{\max}} \sum_{x_h, a_h} \frac{\mu^\dagger_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \\
&\overset{(ii)}{=} \log A \sum_{h=1}^{H} X_h A = X A \log A,
\end{aligned}
$$

where (i) is because $\mu^\dagger_h(a_h|x_h) \log \mu^\dagger_h(a_h|x_h) \le 0$ (recalling that each sequence form $\mu^\dagger_{1:h}(x_h, a_h)$ contains the term $\mu^\dagger_h(a_h|x_h)$), and (ii) uses the balancing property of $\mu^{\star,h}$ (Lemma 37). □

Proof of Lemma 41. By Eq. (6.7) and by the definition of KL divergence, we have

$$
\begin{aligned}
&(X_h A) \mathrm{D}^{\mathrm{kl}}(\mathbb{P}_h^{\mu_{1:h},\star} \| \mathbb{P}_h^{\mu_{1:h-1}\nu_h,\star}) \\
&= (X_h A) \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:h}(x_h, a_h) p^{\star,h}_{1:h}(x_h) \log \Big[\frac{\mu_{1:h}(x_h, a_h) p^{\star,h}_{1:h}(x_h)}{\mu_{1:h-1}(x_{h-1}, a_{h-1}) \nu_h(x_h|a_h) p^{\star,h}_{1:h}(x_h)} \Big] \\
&= \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \frac{\mu_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \log \Big[\frac{\mu_h(a_h|x_h)}{\nu_h(a_h|x_h)} \Big],
\end{aligned}
\tag{E.1}
$$

where the last equality is by Lemma 38. Comparing with the definition of $\mathrm{D}^{\mathrm{bal}}$ as in Eq. (6.5) concludes the proof. □

E.3 Proofs for Section 6.2

E.3.1 Efficient implementation for Update (6.10)

Lemma 123. *Algorithm 23 indeed solves the optimization problem* (6.10)*:*

$$
\mu^{t+1} \leftarrow \arg\min_{\mu \in \Pi_{\max}} \left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\eta} \mathbb{D}(\mu \| \mu^t).
$$

Proof. First, by the sparsity of the loss estimator $\widetilde{\ell}^t$ (cf. (6.9)), the above objective

Algorithm 23 Implementation of Balanced OMD update

```
Require: Current policy $\mu^t$; Trajectory $(x_1^t, a_1^t, \dots, x_H^t, a_H^t)$; learning rate $\eta > 0$;
    Loss vector $\left\{\widetilde{\ell}_h^t(x_h, a_h)\right\}_{h, x_h, a_h}$ that is non-zero only on $(x_h, a_h) = (x_h^t, a_h^t)$.
Set $Z_{H+1}^t \leftarrow 1$.
for $h = H, \dots, 1$ do
    Compute normalization constant
        $$Z_h^t \leftarrow 1 - \mu_h^t(a_h^t|x_h^t) + \mu_h^t(a_h^t|x_h^t) \cdot \exp\left(-\eta\mu_{1:h}^{\star,h}(x_h^t, a_h^t)\widetilde{\ell}_h^t(x_h^t, a_h^t) + \frac{\mu_{1:h}^{\star,h}(x_h^t, a_h^t)\log Z_{h+1}^t}{\mu_{1:h+1}^{\star,h+1}(x_{h+1}^t, a_{h+1}^t)}\right).$$
    Update policy at $x_h^t$:
        $$\mu_h^{t+1}(a_h|x_h^t) \leftarrow \begin{cases} \mu_h^t(a_h|x_h^t) \cdot \exp\left(-\eta\mu_{1:h}^{\star,h}(x_h^t, a_h^t)\widetilde{\ell}_h^t(x_h^t, a_h^t) + \frac{\mu_{1:h}^{\star,h}(x_h^t, a_h^t)\log Z_{h+1}^t}{\mu_{1:h+1}^{\star,h+1}(x_{h+1}^t, a_{h+1}^t)} - \log Z_h^t\right) & \text{if } a_h = a_h^t, \\ \mu_h^t(a_h|x_h^t) \cdot \exp(-\log Z_h^t) & \text{otherwise.} \end{cases}$$
    Set $\mu_h^{t+1}(\cdot|x_h) \leftarrow \mu_h^t(\cdot|x_h)$ for all $x_h \in \mathcal{X}_h \setminus \{x_h^t\}$.
Ensure: Updated policy $\mu^{t+1}$.
```

can be written succinctly as

$$\begin{aligned} & \left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\eta}\mathbb{D}(\mu \| \mu^t) && \text{(E.2)} \\ &= \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) \left[\widetilde{\ell}_h^t\,(x_h, a_h) + \frac{1}{\eta\mu_{1:h}^{\star,h}(x_h, a_h)} \log\frac{\mu_h(a_h|x_h)}{\mu_h^t(a_h|x_h)} \right] \\ &= \sum_{h=1}^{H} \sum_{x_h} \mu_{1:h-1}(x_h) \left[\left\langle \mu_h(\cdot|x_h), \widetilde{\ell}_h^t\,(x_h, \cdot) \right\rangle + \frac{\mathrm{KL}\left(\mu_h(\cdot|x_h) || \mu_h^t(\cdot|x_h)\right)}{\eta\mu_{1:h}^{\star,h}(x_h, a_h)} \right] \\ &= \sum_{h=1}^{H} \{ \mu_{1:h-1}(x_h^t) \left[\mu_h(a_h^t|x_h^t)\widetilde{\ell}_h^t\,\left(x_h^t, a_h^t\right) + \frac{\mathrm{KL}\left(\mu_h(\cdot|x_h) || \mu_h^t(\cdot|x_h)\right)}{\eta\mu_{1:h}^{\star,h}(x_h^t, a_h)} \right] \\ &\quad + \sum_{x_h \neq x_h^t} \mu_{1:h-1}(x_h) \frac{\mathrm{KL}\left(\mu_h(\cdot|x_h) || \mu_h^t(\cdot|x_h)\right)}{\eta\mu_{1:h}^{\star,h}(x_h, a_h)} \}. && \text{(E.3)} \end{aligned}$$

We now show the equivalence by backward induction over $h = H, \dots, 1$. For $h = H$, we can optimize over the H-th layer directly to see

$$\mu_H^{t+1}(a_H|x_H^t) \propto_{a_H} \mu_H^t(a_H|x_H^t) \exp\left\{ -\eta\mu_{1:h}^{\star,h}(x_h^t, a_h)\widetilde{\ell}_H^t(x_H^t, a_H) \right\}$$

$$= \mu_H^t(a_H|x_H^t)\exp\left\{-\eta\widetilde{\ell}_H^t(x_H^t,a_H) - \log Z_H^t\right\},$$

where $Z_H^t > 0$ is the normalization constant. For all non-visited $x_H \neq x_H^t$, by equation (E.3) and non-negativity of KL divergence, the object must be minimized at $\mu_H^{t+1}(\cdot|x_H) = \mu_h^t(\cdot|x_H)$.

If the claim holds from layer $h+1$ to H, consider the h-th layer. Plug in the proved optimizer after layer h, the objective (E.3) can be written as

$$\begin{aligned}
&\sum_{h'=1}^{H}\sum_{x_{h'},a_{h'}}\mu_{1:h'}(x_{h'},a_{h'})\left[\widetilde{\ell}_{h'}^t\left(x_{h'},a_{h'}\right)+\frac{1}{\eta\mu_{1:h'}^{\star,h'}(x_{h'},a_{h'})}\log\frac{\mu_{h'}(a_{h'}|x_{h'})}{\mu_{h'}^t(a_{h'}|x_{h'})}\right]\\
=&\sum_{h'=1}^{H}\sum_{x_{h'}}\mu_{1:h'-1}(x_{h'})\left[\left\langle\mu_{h'}(\cdot|x_{h'}),\widetilde{\ell}_{h'}^t\left(x_{h'},\cdot\right)\right\rangle+\frac{\mathrm{KL}\left(\mu_{h'}(\cdot|x_{h'})||\mu_{h'}^t(\cdot|x_{h'})\right)}{\eta\mu_{1:h'}^{\star,h'}(x_{h'},a_{h'})}\right]\\
=&\sum_{h'=1}^{h}\sum_{x_{h'}}\mu_{1:h'-1}(x_{h'})\left[\left\langle\mu_{h'}(\cdot|x_{h'}),\widetilde{\ell}_{h'}^t\left(x_{h'},\cdot\right)\right\rangle+\frac{\mathrm{KL}\left(\mu_{h'}(\cdot|x_{h'})||\mu_{h'}^t(\cdot|x_{h'})\right)}{\eta\mu_{1:h'}^{\star,h'}(x_{h'},a_{h'})}\right]\\
&+\sum_{h'=h+1}^{H}\left[\frac{\mu_{1:h'}(x_{h'}^t,a_{h'}^t)\log Z_{h'+1}^t}{\eta\mu_{1:h'+1}^{\star,h'+1}(x_{h'+1}^t,a_{h'+1}^t)}-\frac{\mu_{1:h'-1}(x_{h'-1}^t,a_{h'-1}^t)\log Z_{h'}^t}{\eta\mu_{1:h'}^{\star,h'}(x_{h'}^t,a_{h'}^t)}\right]\\
=&\sum_{h'=1}^{h}\sum_{x_{h'}}\mu_{1:h'-1}(x_{h'})\left[\left\langle\mu_{h'}(\cdot|x_{h'}),\widetilde{\ell}_{h'}^t\left(x_{h'},\cdot\right)\right\rangle+\frac{\mathrm{KL}\left(\mu_{h'}(\cdot|x_{h'})||\mu_{h'}^t(\cdot|x_{h'})\right)}{\eta\mu_{1:h'}^{\star,h'}(x_{h'},a_{h'})}\right]\\
&\qquad\qquad-\frac{\mu_{1:h}(x_h^t,a_h^t)\log Z_{h+1}^t}{\eta\mu_{1:h+1}^{\star,h+1}(x_{h+1}^t,a_{h+1}^t)}\\
=&\sum_{h'=1}^{h-1}\sum_{x_{h'}}\mu_{1:h'-1}(x_{h'})\left[\left\langle\mu_{h'}(\cdot|x_{h'}),\widetilde{\ell}_{h'}^t\left(x_{h'},\cdot\right)\right\rangle+\frac{\mathrm{KL}\left(\mu_{h'}(\cdot|x_{h'})||\mu_{h'}^t(\cdot|x_{h'})\right)}{\eta\mu_{1:h'}^{\star,h'}(x_{h'},a_{h'})}\right]\\
&+\mu_{1:h-1}(x_h^t)\left[\mu_h(a_h^t|x_h^t)\left(\widetilde{\ell}_h^t\left(x_h^t,a_h^t\right)-\frac{\log Z_{h+1}^t}{\eta\mu_{1:h+1}^{\star,h+1}(x_{h+1}^t,a_{h+1}^t)}\right)+\frac{\mathrm{KL}\left(\mu_h(\cdot|x_h^t)||\mu_h^t(\cdot|x_h^t)\right)}{\eta\mu_{1:h}^{\star,h}(x_h^t,a_h)}\right]\\
&+\sum_{x_h\neq x_h^t}\mu_{1:h-1}(x_h)\frac{\mathrm{KL}\left(\mu_h(\cdot|x_h)||\mu_h^t(\cdot|x_h)\right)}{\eta\mu_{1:h}^{\star,h}(x_h,a_h)}.
\end{aligned}$$

Thus in the h layer we can optimize by setting

$$\begin{aligned}
&\mu_h^{t+1}(a_h|x_h^t)\\
=&\frac{\mu_h^t(a_h|x_h^t)}{Z_h^t}\exp\left\{-\left[\eta\mu_{1:h}^{\star,h}(x_h^t,a_h)\widetilde{\ell}_h^t(x_h^t,a_h)-\frac{\mu_{1:h}^{\star,h}(x_h^t,a_h)}{\mu_{1:h+1}^{\star,h+1}(x_{h+1}^t,a_{h+1}^t)}\log Z_{h+1}^t\right]\mathbf{1}\left\{a_h=a_h^t\right\}\right\}.
\end{aligned}$$

For all non-visited $x_h \neq x_h^t$, by non-negativity of KL divergence, the object must be minimized at $\mu_h^{t+1}(\cdot|x_h) = \mu_h^t(\cdot|x_h)$. This is exactly the update rule in Algorithm 23.

□

E.3.2 Proof of Theorem 42

Decompose the regret as

$$\mathfrak{R}^T = \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^T \left\langle \mu^t - \mu^\dagger, \ell^t \right\rangle \tag{E.4}$$

$$\leq \underbrace{\sum_{t=1}^T \left\langle \mu^t, \ell^t - \widetilde{\ell}^t \right\rangle}_{\text{BIAS}^1} + \underbrace{\max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^T \left\langle \mu^\dagger, \widetilde{\ell}^t - \ell^t \right\rangle}_{\text{BIAS}^2} + \underbrace{\max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^T \left\langle \mu^t - \mu^\dagger, \widetilde{\ell}^t \right\rangle}_{\text{REGRET}}. \tag{E.5}$$

We now state three lemmas that bound each of the three terms above. Their proofs are presented in Section E.3.4, E.3.5, and E.3.6 respectively. Below, $\iota := \log(3HXA/\delta)$ denotes a log factor.

Lemma 124 (Bound on BIAS^1). *With probability at least $1 - \delta/3$, we have*

$$\text{BIAS}^1 \leq H\sqrt{2T\iota} + \gamma HT.$$

Lemma 125 (Bound on BIAS^2). *With probability at least $1 - \delta/3$, we have*

$$\text{BIAS}^2 \leq XA\iota/\gamma.$$

Lemma 126 (Bound on REGRET). *With probability at least $1 - \delta/3$, we have*

$$\text{REGRET} \leq \frac{XA \log A}{\eta} + \eta H^3 T + \frac{\eta H^2 XA\iota}{\gamma}.$$

Putting the bounds together, we have that with probability at least $1 - \delta$,

$$\mathfrak{R}^T \leq \frac{XA \log A}{\eta} + \eta H^3 T + \frac{\eta H^2 XA\iota}{\gamma} + H\sqrt{2T\iota} + \gamma HT + \frac{XA\iota}{\gamma}.$$

Set $\eta = \sqrt{\frac{XA\log A}{H^3T}}$ and $\gamma = \sqrt{\frac{XA\iota}{TH}}$, we have

$$\mathfrak{R}^T \leq 6\sqrt{XAH^3T\iota} + HXA\iota.$$

Additionally, recall the naive bound $\mathfrak{R}^T \leq HT$ on the regret (which follows as $\langle \mu^t, \ell^t \rangle \in [0, H]$ for any $\mu \in \Pi_{\max}$, $t \in [T]$), we get

$$\mathfrak{R}^T \leq \min\left\{6\sqrt{XAH^3T\iota} + HXA\iota, HT\right\} \leq HT \cdot \min\left\{6\sqrt{XAH\iota/T} + XA\iota/T, 1\right\}.$$

For $T > HXA\iota$, the min above is upper bounded by $7\sqrt{HXA\iota/T}$. For $T \leq HXA\iota$, the min above is upper bounded by $1 \leq 7\sqrt{HXA\iota/T}$. Therefore, we always have

$$\mathfrak{R}^T \leq HT \cdot 7\sqrt{HXA\iota/T} = 7\sqrt{H^3XAT\iota}.$$

This is the desired result. □

The rest of this section is devoted to proving the above three lemmas.

E.3.3 A concentration result

We begin by presenting a useful concentration result. This result is a variant of [Kozuno et al., 2021, Lemma 3] and [Neu, 2015, Lemma 1] suitable to our loss estimator (6.9) where the IX bonus on the denominator depends on (x_h, a_h).

Lemma 127. *For some fixed $h \in [H]$, let $\alpha_h^t(x_h, a_h) \in \left[0, 2\gamma\mu_{1:h}^{\star,h}(x_h, a_h)\right]$ be $\mathcal{F}^{t-1}$-measurable random variable for each $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$. Then with probability $1 - \delta$,*

$$\sum_{t=1}^{T}\sum_{x_h,a_h} \alpha_h^t(x_h, a_h)\left(\widetilde{\ell}_h^t(x_h, a_h) - \ell_h^t(x_h, a_h)\right) \leq \log(1/\delta).$$

Proof. Define the unbiased importance sampling estimator

$$\widehat{\ell}_h^t := \frac{1 - r_h^t}{\mu_{1:h}^t(x_h^t, a_h^t)} \cdot \mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\}.$$

We first have

$$\begin{aligned}
\widetilde{\ell}_h^t\left(x_h,a_h\right)=&\frac{1-r_h^t}{\mu_{1:h}^t(x_h,a_h)+\gamma\mu_{1:h}^{\star,h}(x_h,a_h)}\cdot\mathbf{1}\left\{x_h=x_h^t,a_h=a_h^t\right\}\\
\leq&\frac{1-r_h^t}{\mu_{1:h}^t(x_h,a_h)+\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\left(1-r_h^t\right)}\cdot\mathbf{1}\left\{x_h=x_h^t,a_h=a_h^t\right\}\\
\leq&\frac{1}{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)}\frac{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\left(1-r_h^t\right)\mathbf{1}\left\{x_h=x_h^t,a_h=a_h^t\right\}/\mu_{1:h}^t(x_h,a_h)}{1+\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\left(1-r_h^t\right)\mathbf{1}\left\{x_h=x_h^t,a_h=a_h^t\right\}/\mu_{1:h}^t(x_h,a_h)}\\
=&\frac{1}{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)}\frac{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\widehat{\ell}_h^t(x_h,a_h)}{1+\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\widehat{\ell}_h^t(x_h,a_h)}\\
\overset{(i)}{\leq}&\frac{1}{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)}\log\left(1+2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\widehat{\ell}_h^t(x_h,a_h)\right),
\end{aligned}$$

where (i) is because for any $z\geq 0$, $\frac{z}{1+z/2}\leq\log\left(1+z\right)$.

As a result, we have the following bound on the moment generating function:

$$\begin{aligned}
&\mathbb{E}\left\{\exp\left\{\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\widetilde{\ell}_h^t\left(x_h,a_h\right)\right\}|\mathcal{F}^{t-1}\right\}\\
\leq&\mathbb{E}\left\{\exp\left\{\sum_{x_h,a_h}\frac{\alpha_h^t\left(x_h,a_h\right)}{2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)}\log\left(1+2\gamma\mu_{1:h}^{\star,h}(x_h,a_h)\widehat{\ell}_h^t(x_h,a_h)\right)\right\}|\mathcal{F}^{t-1}\right\}\\
\overset{(i)}{\leq}&\mathbb{E}\left\{\exp\left\{\sum_{x_h,a_h}\log\left(1+\alpha_h^t\left(x_h,a_h\right)\widehat{\ell}_h^t(x_h,a_h)\right)\right\}|\mathcal{F}^{t-1}\right\}\\
=&\mathbb{E}\left\{\prod_{x_h,a_h}\left(1+\alpha_h^t\left(x_h,a_h\right)\widehat{\ell}_h^t(x_h,a_h)\right)|\mathcal{F}^{t-1}\right\}\\
\overset{(ii)}{=}&\mathbb{E}\left\{1+\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\widehat{\ell}_h^t(x_h,a_h)|\mathcal{F}^{t-1}\right\}\\
=&1+\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\ell_h^t(x_h,a_h)\\
\leq&\mathbb{E}\left\{\exp\left\{\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\ell_h^t(x_h,a_h)\right\}|\mathcal{F}^{t-1}\right\},
\end{aligned}$$

where (i) is because $z\log\left(1+z'\right)\leq\log\left(1+zz'\right)$ for any $0\leq z\leq 1$ and $z'>-1$, and (ii) follows from the fact that for any h, at most one of $\widehat{\ell}_h^t(x_h,a_h)$ is non-zero, so the cross terms disappear.

Repeating the above argument,

$$
\begin{aligned}
&\mathbb{E}\left\{\exp\left\{\sum_{t=1}^{T}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\right\}\right\}\\
\leq&\mathbb{E}\{\exp\left\{\sum_{t=1}^{T-1}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\right\}\\
&\qquad\cdot\mathbb{E}\left\{\exp\left\{\sum_{x_h,a_h}\alpha_h^T\left(x_h,a_h\right)\left(\widetilde{\ell}_h^T\left(x_h,a_h\right)-\ell_h^T\left(x_h,a_h\right)\right)\right\}|\mathcal{F}^{T-1}\right\}\}\\
\leq&\mathbb{E}\left\{\exp\left\{\sum_{t=1}^{T-1}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\right\}\right\}\\
\leq&\cdots\leq 1.
\end{aligned}
$$

Therefore, we can apply the Markov inequality and get

$$
\begin{aligned}
&\mathbb{P}\left\{\sum_{t=1}^{T}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)>\log\left(1/\delta\right)\right\}\\
=&\mathbb{P}\left\{\exp\left\{\sum_{t=1}^{T-1}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\right\}>1/\delta\right\}\\
\leq&\delta\cdot\mathbb{E}\left\{\exp\left\{\sum_{t=1}^{T}\sum_{x_h,a_h}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\right\}\right\}\leq\delta.
\end{aligned}
$$

This is the desired result. □

Corollary 128. *We have*

1. *For some fixed $h \in [H]$ and (x_h, a_h), let $\alpha_h^t(x_h, a_h) \in \left[0, 2\gamma\mu_{1:h}^{\star,h}(x_h, a_h)\right]$ be $\mathcal{F}^{t-1}$-measurable random variable. Then with probability $1-\delta$,*

$$\sum_{t=1}^{T}\alpha_h^t\left(x_h,a_h\right)\left(\widetilde{\ell}_h^t\left(x_h,a_h\right)-\ell_h^t\left(x_h,a_h\right)\right)\leq\log\left(1/\delta\right).$$

2. *For some fixed $h \in [H]$ and x_h, let $\alpha_h^t(x_h, a_h) \in \left[0, 2\gamma\mu_{1:h}^{\star,h}(x_h, a_h)\right]$ be $\mathcal{F}^{t-1}$-*

measurable random variable for each $a_h \in \mathcal{A}$. Then with probability $1-\delta$,

$$\sum_{t=1}^{T} \sum_{a_h \in \mathcal{A}} \alpha_h^t\left(x_h, a_h\right)\left(\widetilde{\ell}_h^t\left(x_h, a_h\right)-\ell_h^t\left(x_h, a_h\right)\right) \leq \log (1/\delta) .$$

Proof. For (a), using Lemma 127 with $\left(\alpha_h^t\right)^{\prime}\left(x_h^{\prime}, a_h^{\prime}\right)=\alpha_h^t\left(x_h^{\prime}, a_h^{\prime}\right) \mathbf{1}\left\{x_h^{\prime}=x_h, a_h^{\prime}=a_h\right\}$,

$$\begin{aligned}
& \sum_{t=1}^{T} \alpha_h^t\left(x_h, a_h\right)\left(\widetilde{\ell}_h^t\left(x_h, a_h\right)-\ell_h^t\left(x_h, a_h\right)\right) \\
= & \sum_{t=1}^{T} \sum_{x_h^{\prime}, a_h^{\prime}} \alpha_h^t\left(x_h, a_h\right) \mathbf{1}\left\{x_h^{\prime}=x_h, a_h^{\prime}=a_h\right\}\left[\widetilde{\ell}^t(x_h^{\prime}, a_h^{\prime})-\ell^t(x_h^{\prime}, a_h^{\prime})\right] \leq \log (1/\delta) .
\end{aligned}$$

Claim (b) can proved similarly. □

E.3.4 Proof of Lemma 124

We further decompose BIAS^1 to two terms by

$$\mathrm{BIAS}^1=\sum_{t=1}^{T}\left\langle\mu^t, \ell^t-\widetilde{\ell}^t\right\rangle=\underbrace{\sum_{t=1}^{T}\left\langle\mu^t, \ell^t-\mathbb{E}\left\{\widetilde{\ell}^t | \mathcal{F}^{t-1}\right\}\right\rangle}_{(A)}+\underbrace{\sum_{t=1}^{T}\left\langle\mu^t, \mathbb{E}\left\{\widetilde{\ell}^t | \mathcal{F}^{t-1}\right\}-\widetilde{\ell}^t\right\rangle}_{(B)}.$$

To bound (A), plug in the definition of loss estimator,

$$\begin{aligned}
& \sum_{t=1}^{T}\left\langle\mu^t, \ell^t-\mathbb{E}\left\{\widetilde{\ell}^t | \mathcal{F}^{t-1}\right\}\right\rangle \\
= & \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}^t(x_h, a_h)\left[\ell_h^t(x_h, a_h)-\frac{\mu_{1:h}^t(x_h, a_h) \ell_h^t(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h)+\gamma \mu_{1:h}^{\star, h}(x_h, a_h)}\right] \\
= & \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}^t(x_h, a_h) \ell_h^t(x_h, a_h)\left[\frac{\gamma \mu_{1:h}^{\star, h}(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h)+\gamma \mu_{1:h}^{\star, h}(x_h, a_h)}\right] \\
\leq & \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h, a_h} \gamma \mu_{1:h}^{\star, h}(x_h, a_h) \ell_h^t(x_h, a_h) \\
\overset{(i)}{\leq} & \sum_{t=1}^{T} \sum_{h=1}^{H} \gamma=\gamma H T,
\end{aligned}$$

where (i) is by using Corollary 111 with policy $\mu^{\star,h}$ for each layer h.

To bound (B), first notice

$$\begin{aligned}\left\langle \mu^t, \widetilde{\ell}^t \right\rangle =& \sum_{h=1}^{H} \sum_{x_h,a_h} \mu_{1:h}^t(x_h,a_h) \frac{(1-r_h^t)\,\mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\}}{\mu_{1:h}^t(x_h,a_h) + \gamma \mu_{1:h}^{\star,h}(x_h,a_h)} \\ \leq& \sum_{h=1}^{H} \sum_{x_h,a_h} \mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\} = \sum_{h=1}^{H} 1 = H.\end{aligned}$$

Then by Azuma-Hoeffding, with probability at least $1-\delta/3$,

$$\sum_{t=1}^{T} \left\langle \mu^t, \mathbb{E}\left\{\widetilde{\ell}^t | \mathcal{F}^{t-1}\right\} - \widetilde{\ell}^t \right\rangle \leq H\sqrt{2T\log(3/\delta)} \leq H\sqrt{2T\iota}.$$

Combining the bounds for (A) and (B) gives the desired result. □

E.3.5 Proof of Lemma 125

We have

$$\begin{aligned}\mathrm{BIAS}^2 =& \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \left\langle \mu^\dagger, \widetilde{\ell}^t - \ell^t \right\rangle \\ =& \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h,a_h} \mu_{1:h}^\dagger(x_h,a_h) \left[\widetilde{\ell}_h^t(x_h,a_h) - \ell_h^t(x_h,a_h)\right] \\ =& \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h,a_h} \frac{\mu_{1:h}^\dagger(x_h,a_h)}{\gamma\mu_{1:h}^{\star,h}(x_h,a_h)} \gamma\mu_{1:h}^{\star,h}(x_h,a_h) \left[\widetilde{\ell}_h^t(x_h,a_h) - \ell_h^t(x_h,a_h)\right] \\ =& \max_{\mu^\dagger \in \Pi_{\max}} \sum_{h=1}^{H} \sum_{x_h,a_h} \frac{\mu_{1:h}^\dagger(x_h,a_h)}{\gamma\mu_{1:h}^{\star,h}(x_h,a_h)} \sum_{t=1}^{T} \gamma\mu_{1:h}^{\star,h}(x_h,a_h) \left[\widetilde{\ell}_h^t(x_h,a_h) - \ell_h^t(x_h,a_h)\right] \\ \overset{(i)}{\leq}& \frac{\log(XA/\delta)}{\gamma} \sum_{h=1}^{H} \max_{\mu^\dagger \in \Pi_{\max}} \sum_{x_h,a_h} \frac{\mu_{1:h}^\dagger(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)} \\ \overset{(ii)}{\leq}& \frac{\iota}{\gamma} \sum_{h=1}^{H} X_h A = XA\iota/\gamma,\end{aligned}$$

where (i) is by applying Corollary 128 for each (x_h,a_h) pair and taking union bound, and (ii) is by Lemma 37. □

E.3.6 Proof of Lemma 126

We begin by stating the following lemma, which roughly speaking relates the task of bounding the regret to bounding the term $\left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\eta\mu^{\star,1}_{1:1}(x^t_1, a_1)} \log Z^t_1$.

Lemma 129. *For any policy $\mu \in \Pi_{\max}$,*

$$\mathbb{D}(\mu \| \mu^{t+1}) - \mathbb{D}(\mu \| \mu^t) = \eta \left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\mu^{\star,1}_{1:1}(x^t_1, a_1)} \log Z^t_1.$$

Proof. By definition of $\mathbb{D}$ and the conditional form update rule in Algorithm 10,

$$\begin{aligned}
&\mathbb{D}(\mu \| \mu^{t+1}) - \mathbb{D}(\mu \| \mu^t) \\
=&\sum_{h=1}^{H} \sum_{x_h, a_h} \frac{\mu_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \log \frac{\mu^t_h(a_h|x_h)}{\mu^{t+1}_h(a_h|x_h)} \\
=&\sum_{h=1}^{H} \sum_{a_h} \frac{\mu_{1:h}(x^t_h, a_h)}{\mu^{\star,h}_{1:h}(x^t_h, a_h)} \log \frac{\mu^t_h(a_h|x^t_h)}{\mu^{t+1}_h(a_h|x^t_h)} \\
=&\sum_{h=1}^{H} \frac{\mu_{1:h}(x^t_h, a^t_h)}{\mu^{\star,h}_{1:h}(x^t_h, a^t_h)} \left[\eta \mu^{\star,h}_{1:h}(x^t_h, a^t_h) \widetilde{\ell}^t_h - \frac{\mu^{\star,h}_{1:h}(x^t_h, a^t_h)}{\mu^{\star,h+1}_{1:h+1}(x^t_{h+1}, a^t_{h+1})} \log Z^t_{h+1} \right] \\
&\qquad + \sum_{h=1}^{H} \sum_{a_h} \frac{\mu_{1:h}(x^t_h, a_h)}{\mu^{\star,h}_{1:h}(x^t_h, a_h)} \log Z^t_h \\
=&\eta \sum_{h=1}^{H} \mu_{1:h}(x^t_h, a^t_h) \widetilde{\ell}^t_h(x^t_h, a^t_h) - \sum_{h=1}^{H} \frac{\mu_{1:h}(x^t_h, a^t_h)}{\mu^{\star,h+1}_{1:h+1}(x^t_{h+1}, a^t_{h+1})} \log Z^t_{h+1} \\
&\qquad + \sum_{h=1}^{H} \frac{\mu_{1:h-1}(x^t_{h-1}, a^t_{h-1})}{\mu^{\star,h}_{1:h}(x^t_h, a^t_h)} \log Z^t_h \\
=&\eta \left\langle \mu, \widetilde{\ell}^t \right\rangle + \frac{1}{\mu^{\star,1}_{1:1}(x^t_1, a_1)} \log Z^t_1.
\end{aligned}$$

□

Additional notation We introduce the following notation for convenience throughout the rest of this subsection. Define

$$\beta^t_h := \eta \mu^{\star,h}_{1:h}(x^t_h, a^t_h).$$

For simplicity, when there is no confusion, we write

$$\mu_h^t := \mu_h^t(a_h^t|x_h^t), \quad \mu_{h:h'}^t := \prod_{h''=h}^{h'} \mu_{h''}^t,$$

and

$$\widetilde{\ell}_h^t := \widetilde{\ell}_h^t\left(x_h^t, a_h^t\right) = \frac{1 - r_h^t}{\mu_{1:h}^t(x_h^t, a_h^t) + \gamma\mu_{1:h}^\star(x_h^t, a_h^t)}.$$

Define the normalized log-partition function as

$$\Xi_h^t := \frac{1}{\beta_h^t}\log Z_h^t = \frac{1}{\beta_h^t}\log\left(1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t\left(\Xi_{h+1}^t - \widetilde{\ell}_h^t\right)\right]\right).$$

Note that this value can be seen as an H-variate function of the loss estimator $\left\{\widetilde{\ell}_h^t\right\}_{h\in[H]}$. To make this dependence more clear, for any $\widetilde{\ell} \in [0,\infty)^H$, we define the function $\{\Xi_h^t\left(\cdot\right)\}_{h=1}^H$ recursively by

$$\Xi_h^t\left(\widetilde{\ell}_{h:H}\right) := \begin{cases} \log\left(1 - \mu_h^t + \mu_h^t \exp\left[-\beta_h^t\widetilde{\ell}_h\right]\right)/\beta_h^t & \text{if } \; h = H, \\ \log\left(1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t\left(\Xi_{h+1}\left(\widetilde{\ell}_{h+1:H}\right) - \widetilde{\ell}_h\right)\right]\right)/\beta_h^t & \text{otherwise.}\end{cases}$$

We also overload the notation by $\Xi_h^t\left(\widetilde{\ell}\right) = \Xi_h^t\left(\widetilde{\ell}_{h:H}\right)$ and $\Xi_h^t = \Xi_h^t\left(\widetilde{\ell}^t\right)$, where $\widetilde{\ell}^t$ is the actual loss estimator. Note that, importantly, $\Xi_h^t(\widetilde{\ell}_{h:H})$ has a *compositional structure*: It is a function of $\widetilde{\ell}_h$ (h-th entry of the loss) and Ξ_{h+1}^t (which is itself a function of $\widetilde{\ell}_{h+1:H}$). This compositional structure is key to proving bounds on its gradients and Hessians.

The rest of this subsection is organized as follows. In Section E.3.6, we bound the gradients and Hessians of the function $\Xi_1^t(\cdot)$ in an entry-wise fashion, and then use the Mean-Value Theorem to give a bound on $\Xi_1^t = \Xi_1^t(\widetilde{\ell}^t)$ (Lemma 133). We then combine this result with Lemma 129 to prove the main lemma that bounds REGRET (Section E.3.6).

Bounding Ξ_1^t

Lemma 130. *For $\widetilde{\ell} \in [0,\infty)^H$ and any $h \in [H]$, $\Xi_h^t\left(\widetilde{\ell}\right) \leq 0$. Furthermore, $\Xi_h^t(0) = 0$.*

Proof. We show the first claim by backward induction. For $h = H$,

$$\Xi_H^t\left(\widetilde{\ell}_H\right) = \log\left(1 - \mu_H^t + \mu_H^t \exp\left[-\beta_H^t \widetilde{\ell}_H\right]\right)/\beta_H^t \leq \log\left(1 - \mu_H^t + \mu_H^t\right)/\beta_H^t \leq 0,$$

because $\widetilde{\ell}_H^t \geq 0$.

Assume $\Xi_{h+1}^t\left(\widetilde{\ell}\right) \leq 0$, then for the previous step h,

$$\begin{aligned}\Xi_h^t\left(\widetilde{\ell}_{h:H}\right) &= \log\left(1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t\left(\Xi_{h+1}^t\left(\widetilde{\ell}_{h+1:H}\right) - \widetilde{\ell}_h\right)\right]\right)/\beta_h^t \\ &\leq \log\left(1 - \mu_h^t + \mu_h^t\right)/\beta_h^t \leq 0.\end{aligned}$$

The second claim follows as all inequalities become equalities at $\widetilde{\ell} = 0$. □

Lemma 131 (Bounds on first derivatives)**.** *For $\widetilde{\ell} \in [0,1]^H$ and any $h \in [H]$, the derivatives are bounded by*

$$0 \leq \frac{\partial \Xi_h^t}{\partial \Xi_{h+1}^t} \leq \mu_h^t \quad \text{and} \quad -\mu_h^t \leq \frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_h} \leq 0.$$

Furthermore,

$$\left.\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_{h'}}\right|_{\widetilde{\ell}=0} = \begin{cases} -\mu_{h:h'}^t & \text{if } h' \geq h, \\ 0 & \text{otherwise.}\end{cases}$$

Proof. By chain rule and the compositional structure of the functions $\Xi_h^t(\cdot)$, for any $h' \geq h$,

$$\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_{h'}} = \frac{\partial \Xi_h^t}{\partial \Xi_{h'}^t} \cdot \frac{\partial \Xi_{h'}^t}{\partial \widetilde{\ell}_{h'}} = \left(\prod_{h''=h}^{h'-1} \frac{\partial \Xi_{h''}^t}{\partial \Xi_{h''+1}^t}\right) \cdot \frac{\partial \Xi_{h'}^t}{\partial \widetilde{\ell}_{h'}}.$$

For any h, the derivatives are bounded by

$$\frac{\partial \Xi_h^t}{\partial \Xi_{h+1}^t} = \frac{\mu_h^t \exp\left[\beta_h^t\left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]}{1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t\left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]} \in \left[0, \mu_h^t\right],$$

$$\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_h} = -\frac{\mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]}{1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]} \in \left[-\mu_h^t, 0\right].$$

The inequalities hold because the function $f\left(z\right) = \frac{\mu_h^t z}{1-\mu_h^t+\mu_h^t z} = 1 - \frac{1-\mu_h^t}{1-\mu_h^t+\mu_h^t z}$ is increasing on $z \in [0,1]$, and $\exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right] \in [0,1]$ by Lemma 130.

Putting them together, at $\widetilde{\ell} = 0$, the derivative is just $\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_{h'}}\Big|_{\widetilde{\ell}^t=0} = -\mu_{h:h'}^t$ if $h' \geq h$. If $h' < h$, since Ξ_h^t only depends on loss in the later layers, $\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_{h'}}|_{\widetilde{\ell}^t=0} = 0$. □

Lemma 132 (Bounds on second derivatives)**.** *For $\widetilde{\ell} \in [0,1]^H$ and any $h \in [H]$, if $h' \geq h$ and $h'' \geq h$, the second-order derivatives are bounded by*

$$\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} \leq \sum_{h'''=h}^{\min\{h',h''\}} \beta_{h'''}^t \mu_{h:h'}^t \mu_{h'''+1:h''}^t = \sum_{h'''=h}^{\min\{h',h''\}} \beta_{h'''}^t \mu_{h:h'''}^t \mu_{h'''+1:h'}^t \mu_{h'''+1:h''}^t.$$

Otherwise $\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} = 0.$

Proof. By symmetry of the second derivatives and the right-hand side with respect to h' and h'', it suffices to prove the claim for $h'' \geq h'$ only.

By chain rule and the compositional structure of the functions $\Xi_h^t(\cdot)$,

$$\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} = \frac{\partial^2 \Xi_h^t}{\partial \Xi_{h'}^t \partial \widetilde{\ell}_{h''}} \cdot \frac{\partial \Xi_{h'}^t}{\partial \widetilde{\ell}_{h'}} + \frac{\partial \Xi_h^t}{\partial \Xi_{h'}^t} \cdot \frac{\partial^2 \Xi_{h'}^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}}.$$

If $h'' = h' = h$,

$$\begin{aligned} \frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_h^2} =& \beta_h^t \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right] \frac{1 - \mu_h^t}{\left\{1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]\right\}^2} \\ \leq& \beta_h^t \mu_h^t. \end{aligned}$$

If $h' = h, h'' > h$,

$$\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_h \partial \widetilde{\ell}_{h''}} = -\frac{(1-\mu_h^t)\beta_h^t \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]}{\left(1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t \left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]\right)^2} \cdot \frac{\partial \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h''}} \leq \beta_h^t \mu_{h:h''}^t.$$

If $h < h' < h''$, we can compute the Hessian by induction. Notice once $h' > h$ we have

$$\frac{\partial \Xi_h^t}{\partial \widetilde{\ell}_{h'}} = \frac{\partial \Xi_h^t}{\partial \Xi_{h+1}^t} \cdot \frac{\partial \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'}}.$$

Take second derivative,

$$\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} = \underbrace{\frac{\partial \Xi_h^t}{\partial \Xi_{h+1}^t} \cdot \frac{\partial^2 \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}}}_{(i)} + \underbrace{\frac{\partial^2 \Xi_h^t}{\partial \Xi_{h+1}^t \partial \widetilde{\ell}_{h''}} \cdot \frac{\partial \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'}}}_{(ii)}.$$

We first bound the second term,

$$\begin{aligned} (ii) &= \frac{(1-\mu_h^t)\beta_h^t \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]}{\left(1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]\right)^2} \cdot \frac{\partial \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h''}} \cdot \frac{\partial \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'}} \\ &\le \beta_h^t \mu_h^t \cdot \mu_{h+1:h''}^t \cdot \mu_{h+1:h'}^t \\ &\le \beta_h^t \mu_{h:h'}^t \mu_{h+1:h''}^t. \end{aligned}$$

The first term can be simplified to

$$(i) \le \frac{\mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]}{1 - \mu_h^t + \mu_h^t \exp\left[\beta_h^t \left(\Xi_{h+1}^t\left(\widetilde{\ell}\right) - \widetilde{\ell}_h\right)\right]} \frac{\partial^2 \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} \le \mu_h^t \frac{\partial^2 \Xi_{h+1}^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}}.$$

Now plug in $\frac{\partial^2 \Xi_{h'}^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} \le \beta_{h'}^t \mu_{h':h''}^t$ and backward induction from h' to h gives:

$$\frac{\partial^2 \Xi_h^t}{\partial \widetilde{\ell}_{h'} \partial \widetilde{\ell}_{h''}} \le \sum_{h'''=h}^{h'} \beta_{h'''}^t \mu_{h:h'}^t \mu_{h'''+1:h''}^t.$$

We can check this expression is also correct for the above special cases when $h' = h$. The second claim holds because Ξ_h^t only depends on loss in the later layers. □

Lemma 133 (Bound on Ξ_1^t)**.** *We have*

$$\Xi_1^t \le -\left\langle \mu^t, \widetilde{\ell}^t \right\rangle + \frac{\eta H}{2} \sum_{h=1}^{H} \left(\sum_{h'=h}^{H} \sum_{x_{h'}, a_{h'}} \mu_{1:h}^{\star,h}\left(x_{h'}, a_{h'}\right) \mu_{h+1:h'}^t\left(x_{h'}, a_{h'}\right) \widetilde{\ell}_{h'}^t\left(x_{h'}, a_{h'}\right) \right).$$

Proof. We apply the Mean-value Theorem to function $\Xi_1^t\left(\widetilde{\ell}\right)$ at $\widetilde{\ell} = 0$,

$$\Xi_1^t = \Xi_1^t\left(\widetilde{\ell}^t\right) = \Xi_1^t(0) + \left\langle \nabla_{\widetilde{\ell}} \Xi_1^t\big|_{\widetilde{\ell}=0}, \widetilde{\ell}^t \right\rangle + \frac{1}{2}\left\langle \nabla_{\widetilde{\ell}}^2\, \Xi_1^t\big|_{\widetilde{\ell}=\xi^t} \widetilde{\ell}^t, \widetilde{\ell}^t \right\rangle,$$

where ξ^t lies on the line segment between 0 and $\widetilde{\ell}^t$.

By Lemma 130, the initial term is just zero. By Lemma 131, the first-order term is just $-\left\langle \mu^t, \widetilde{\ell}^t \right\rangle$.

It thus remains to bound the second-order term. Applying the entry-wise upper bounds in Lemma 132 at $h = 1$ (which hold uniformly at all nonnegative loss values, including ξ^t), we have

$$\begin{aligned}
\left\langle \nabla_{\widetilde{\ell}}^2\, \Xi_1^t\big|_{\widetilde{\ell}=\xi^t} \widetilde{\ell}^t, \widetilde{\ell}^t \right\rangle =& \sum_{h=1}^H \sum_{h'=1}^H \frac{\partial^2 \Xi_1^t}{\partial \widetilde{\ell}_h \partial \widetilde{\ell}_{h'}}\bigg|_{\widetilde{\ell}=\xi^t} \widetilde{\ell}_h^t \widetilde{\ell}_{h'}^t \\
\overset{(i)}{\leq}& \sum_{h=1}^H \sum_{h'=1}^H \sum_{h''=1}^{\min\{h,h'\}} \beta_{h''}^t \mu_{1:h}^t \mu_{h''+1:h'}^t \widetilde{\ell}_h^t \widetilde{\ell}_{h'}^t \\
=& \sum_{h=1}^H \mu_{1:h}^t \widetilde{\ell}_h^t \sum_{h'=1}^H \sum_{h''=1}^{\min\{h,h'\}} \beta_{h''}^t \mu_{h''+1:h'}^t \widetilde{\ell}_{h'}^t \\
\overset{(ii)}{\leq}& H \max_{h\in[H]} \sum_{h'=1}^H \sum_{h''=1}^{\min\{h,h'\}} \beta_{h''}^t \mu_{h''+1:h'}^t \widetilde{\ell}_{h'}^t \\
=& H \sum_{h'=1}^H \sum_{h''=1}^{h'} \beta_{h''}^t \mu_{h''+1:h'}^t \widetilde{\ell}_{h'}^t \\
=& H \sum_{h''=1}^H \sum_{h'=h''}^{H} \beta_{h''}^t \mu_{h''+1:h'}^t \widetilde{\ell}_{h'}^t \\
=& \eta H \sum_{h''=1}^H \left(\sum_{h'=h''}^H \mu_{1:h''}^{\star,h''}\left(x_{h'}^t, a_{h'}^t\right) \mu_{h''+1:h'}^t \widetilde{\ell}_{h'}^t \right) \\
\overset{(iii)}{=}& \eta H \sum_{h''=1}^H \left(\sum_{h'=h''}^H \sum_{x_{h'}, a_{h'}} \mu_{1:h''}^{\star,h''}(x_{h'}, a_{h'})\, \mu_{h''+1:h'}^t(x_{h'}, a_{h'})\, \widetilde{\ell}_{h'}^t(x_{h'}, a_{h'}) \right),
\end{aligned}$$

where (i) is by Lemma 132; (ii) follows from the bound

$$\sum_{h=1}^H \mu_{1:h}^t \widetilde{\ell}_h^t = \sum_{h=1}^H \mu_{1:h}^t \cdot \frac{1 - r_h^t}{\mu_{1:h}^t + \gamma \mu_{1:h}^{\star,h}} \leq H;$$

and (iii) is because $\widetilde{\ell}^t_{h'}(x_{h'}, a_{h'}) = 0$ at all $(x_{h'}, a_{h'}) \neq (x^t_{h'}, a^t_{h'})$. □

Lemma 134. *With probability at least* $1 - \delta/3$,

$$\sum_{t=1}^{T} \Xi^t_1 \leq - \sum_{t=1}^{T} \left\langle \mu^t, \widetilde{\ell}^t \right\rangle + \eta H^3 T + \frac{\eta X A H^2 \iota}{\gamma},$$

where $\iota := \log(H/\delta)$.

Proof. Using Lemma 133 and take the summation with respect to $t \in [T]$ we have

$$\sum_{t=1}^{T} \Xi^t_1 \leq - \sum_{t=1}^{T} \left\langle \mu^t, \widetilde{\ell}^t \right\rangle + \frac{\eta H}{2} \sum_{h=1}^{H} \sum_{h\prime=h}^{H} \sum_{t=1}^{T} \underbrace{\sum_{x_{h\prime}, a_{h\prime}} \mu^{\star,h}_{1:h} \left(x_{h\prime}, a_{h\prime}\right) \mu^t_{h+1:h\prime} \left(x_{h\prime}, a_{h\prime}\right) \widetilde{\ell}^t_{h\prime} \left(x_{h\prime}, a_{h\prime}\right)}_{:=\Delta^t_{h,h\prime}}. \tag{E.6}$$

Observe that the random variables $\Delta^t_{h,h\prime}$ satisfy the following:

- $\Delta^t_{h,h\prime} \leq X_{h'} A/\gamma$ almost surely:

$$\begin{aligned}\Delta^t_{h,h\prime} &= \sum_{x_{h\prime}, a_{h\prime}} \mu^{\star,h}_{1:h} \left(x_{h'}, a_{h'}\right) \mu^t_{h+1:h\prime} \left(x_{h\prime}, a_{h\prime}\right) \frac{(1 - r^t_{h'}) \mathbf{1} \left\{x_{h\prime} = x^t_{h'}, a_{h\prime} = a^t_{h'}\right\}}{\mu^t_{1:h\prime} \left(x_{h\prime}, a_{h\prime}\right) + \gamma \mu^{\star,h'}_{1:h'} \left(x_{h\prime}, a_{h\prime}\right)} \\ &\leq \frac{1}{\gamma} \sum_{x_{h\prime}, a_{h\prime}} \frac{\mu^{\star,h}_{1:h} \left(x_{h'}, a_{h'}\right) \mu^t_{h+1:h\prime} \left(x_{h\prime}, a_{h\prime}\right)}{\mu^{\star,h'}_{1:h'} \left(x_{h\prime}, a_{h\prime}\right)} \overset{(i)}{\leq} \frac{X_{h'} A}{\gamma},\end{aligned}$$

  where (i) is by using Lemma 37 with the mixture of $\mu^{\star,h}$ and μ^t.

- $\mathbb{E}[\Delta^t_{h,h\prime} | \mathcal{F}_{t-1}] \leq 1$, where $\mathcal{F}_{t-1}$ is the σ-algebra containing all information after iteration $t - 1$:

$$\mathbb{E}[\Delta^t_{h,h\prime} | \mathcal{F}_{t-1}] = \sum_{x_{h\prime}, a_{h\prime}} \mu^{\star,h}_{1:h} \left(x_{h'}, a_{h'}\right) \mu^t_{h+1:h\prime} \left(x_{h\prime}, a_{h\prime}\right) \ell^t_{h\prime} \left(x_{h\prime}, a_{h\prime}\right) \overset{(i)}{\leq} 1,$$

  where (i) is by using Corollary 111 with the mixture policy of $\mu^{\star,h}$ and μ^t.

- The conditional variance $\mathbb{E}[(\Delta^t_{h,h\prime})^2 | \mathcal{F}_{t-1}]$ can be bounded as

$$\mathbb{E}[(\Delta^t_{h,h\prime})^2 | \mathcal{F}_{t-1}]$$

$$
\begin{aligned}
&\overset{(i)}{=} \sum_{x_{h'},a_{h'}} \left[\left(\mu_{1:h}^{\star,h}\left(x_{h'},a_{h'}\right) \mu_{h+1:h'}^{t}\left(x_{h'},a_{h'}\right) \frac{\left(1-r_{h'}^{t}\right) \mathbf{1}\left\{x_{h'}=x_{h'}^{t}, a_{h'}=a_{h'}^{t}\right\}}{\mu_{1:h'}^{t}\left(x_{h'},a_{h'}\right)+\gamma \mu_{1:h'}^{\star,h'}\left(x_{h'},a_{h'}\right)} \right)^{2} \right] \\
&\leq \sum_{x_{h'},a_{h'}} \left(\frac{\mu_{1:h}^{\star,h}\left(x_{h'},a_{h'}\right) \mu_{h+1:h'}^{t}\left(x_{h'},a_{h'}\right)}{\mu_{1:h'}^{t}\left(x_{h'},a_{h'}\right)+\gamma \mu_{1:h'}^{\star,h'}\left(x_{h'},a_{h'}\right)} \right)^{2} \mu_{1:h'}^{t}\left(x_{h'},a_{h'}\right) \\
&\leq \frac{1}{\gamma} \sum_{x_{h'},a_{h'}} \frac{\mu_{1:h}^{\star,h}\left(x_{h'},a_{h'}\right) \mu_{h+1:h'}^{t}\left(x_{h'},a_{h'}\right)}{\mu_{1:h'}^{\star,h'}\left(x_{h'},a_{h'}\right)} \overset{(ii)}{\leq} \frac{X_{h'}A}{\gamma},
\end{aligned}
$$

where (i) follows from the fact that for any h, at most one of indicators is non-zero, so the cross terms disappear and (ii) is using Corollary 111 with the mixture policy of $\mu^{\star,h}$ and μ^{t}.

Therefore, we can apply Freedman's inequality (Lemma 119) and union bound to get that, with probability at least $1-\delta/3$, for some fixed $\lambda_{h,h'} \in (0, \gamma/X_{h'}A]$, the following holds simultaneously for all h, h':

$$
\sum_{t=1}^{T} \Delta_{h,h'}^{t} \leq \frac{\lambda_{h,h'} X_{h'} A T}{\gamma} + \frac{2\log(H/\delta)}{\lambda_{h,h'}} + T,
$$

Take $\lambda_{h,h'} = \gamma/X_{h'}A$, we have

$$
\sum_{t=1}^{T} \Delta_{h,h'}^{t} \leq \frac{X_{h'}A \cdot 2\log(H/\delta)}{\gamma} + 2T.
$$

Plug into equation (E.6), we have

$$
\sum_{t=1}^{T} \Xi_{1}^{t} \leq -\sum_{t=1}^{T} \left\langle \mu^{t}, \widetilde{\ell}^{t} \right\rangle + \eta H^{3} T + \frac{\eta H^{2} X A \iota}{\gamma},
$$

where $\iota := \log(H/\delta)$ is a log factor. □

Proof of main lemma

By Lemma 129, for any policy $\mu^{\dagger} \in \Pi_{\max}$,

$$
\frac{1}{\eta}\left(\mathbb{D}(\mu^{\dagger} \| \mu^{t+1}) - \mathbb{D}(\mu^{\dagger} \| \mu^{t})\right) = \left\langle \mu^{\dagger}, \widetilde{\ell}^{t} \right\rangle + \Xi_{1}^{t}.
$$

Taking the summation w.r.t. $t \in [T]$ and using Lemma 134, we have with probability at least $1-\delta/3$, the following holds simultaneously over all $\mu^\dagger \in \Pi_{\max}$:

$$\begin{aligned}\frac{1}{\eta}\left(\mathbb{D}(\mu^\dagger \| \mu^T) - \mathbb{D}(\mu^\dagger \| \mu^1)\right) =& \sum_{t=1}^T \left\langle \mu^\dagger, \widetilde{\ell}^t \right\rangle + \sum_{t=1}^T \Xi_1^t \\ \le& \sum_{t=1}^T \left\langle \mu^\dagger - \mu^t, \widetilde{\ell}^t \right\rangle + \eta H^3 T + \frac{\eta H^2 X A \iota}{\gamma}.\end{aligned}$$

Rerranging the terms we have

$$\begin{aligned}\max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^T \left\langle \mu^t - \mu^\dagger, \widetilde{\ell}^t \right\rangle \le& \max_{\mu^\dagger \in \Pi_{\max}} \frac{1}{\eta}\left(\mathbb{D}(\mu^\dagger \| \mu^1) - \mathbb{D}(\mu^\dagger \| \mu^T)\right) + \eta H^3 T + \frac{\eta H^2 X A \iota}{\gamma} \\ \le& \max_{\mu^\dagger \in \Pi_{\max}} \frac{1}{\eta} \mathbb{D}(\mu^\dagger \| \mu^1) + \eta H^3 T + \frac{\eta H^2 X A \iota}{\gamma} \\ \le& \frac{XA \log A}{\eta} + \eta H^3 T + \frac{\eta H^2 X A \iota}{\gamma},\end{aligned}$$

where the last inequality above follows by recalling that μ^1 is taken to be the uniform policy ($\mu_h^1(a_h|x_h) = 1/A$ for all (h, x_h, a_h)) in Algorithm 10, and applying the bound on the balanced dilated KL (Lemma 40). This proves Lemma 126.

E.4 Proofs for Section 6.3

E.4.1 Counterfactual regret decomposition

Define the immediate counterfactual regret at any $x_h \in \mathcal{X}_h$, $h \in [H]$ as

$$\mathfrak{R}_h^{\mathrm{imm},T}(x_h) = \max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^T \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), L_h^t(x_h, \cdot) \right\rangle, \tag{E.7}$$

where $L_h^t(\cdot,\cdot)$ is the counterfactual loss function defined in (6.11):

$$L_h^t(x_h, a_h) := \ell_h^t(x_h, a_h) + \sum_{h'=h+1}^H \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h) \times \mathcal{A}} \mu_{(h+1):h'}^t(x_{h'}, a_{h'}) \ell_{h'}^t(x_{h'}, a_{h'}).$$

Lemma 135 (Counterfactual regret decomposition)**.** *We have* $\widetilde{\mathfrak{R}}^T \le \sum_{h=1}^H \mathfrak{R}_h^T$*, where*

$$\mathfrak{R}_h^T := \sum_{x_1\in\mathcal{X}_1} \max_{a_1\in\mathcal{A}} \cdots \sum_{x_{h-1}\in\mathcal{C}(x_{h-2},a_{h-2})} \max_{a_{h-1}\in\mathcal{A}} \sum_{x_h\in\mathcal{C}(x_{h-1},a_{h-1})} \mathfrak{R}_h^{\mathrm{imm},T}(x_h),$$
$$= \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1},a_{h-1}) \cdot \mathfrak{R}_h^{\mathrm{imm},T}(x_h).$$

Proof. The bound $\widetilde{\mathfrak{R}}^T \le \sum_{h=1}^H \mathfrak{R}_h^T$ with the sum-max form expression for $\widetilde{\mathfrak{R}}_h^T$ has already implicitly appeared in the proof of [Zinkevich et al., 2007, Theorem 3], albeit with their slightly different formulation of extensive-form games (turn-based games with reward only in the last round). For completeness, here we provide a proof under our formulation.

We first show the bound with the μ form expression for $\mathfrak{R}_h^T$, which basically follows by a performance decomposition argument. We have

$$\begin{aligned}\widetilde{\mathfrak{R}}^T &= \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \left\langle \mu^t - \mu^\dagger, \ell^t \right\rangle \\ &= \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \sum_{h=1}^H \left\langle \mu_{1:h-1}^\dagger \mu_{h:H}^t - \mu_{1:h}^\dagger \mu_{h+1:H}^t, \ell^t \right\rangle \\ &\le \sum_{h=1}^H \underbrace{\max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \left\langle \mu_{1:h-1}^\dagger \mu_{h:H}^t - \mu_{1:h}^\dagger \mu_{h+1:H}^t, \ell^t \right\rangle}_{:=\mathfrak{R}_h^T}.\end{aligned}$$

Note that each term $\mathfrak{R}_h^T$ measures the performance difference between $\mu_{1:h-1}^\dagger \mu_{h:H}^t$ and $\mu_{1:h}^\dagger \mu_{h+1:H}^t$:

$$\begin{aligned}\mathfrak{R}_h^T &= \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \mathbb{E}_{s_h\sim\mu_{1:h-1}^\dagger\times\nu^t}\left[\mathbb{E}_{a_h\sim\mu^t(\cdot|x_h)}\left[\sum_{h'=1}^H r_{h'}\right] - \mathbb{E}_{a_h\sim\mu^\dagger(\cdot|x_h)}\left[\sum_{h'=1}^H r_{h'}\right]\right] \\ &\overset{(i)}{=} \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \mathbb{E}_{s_h\sim\mu_{1:h-1}^\dagger\times\nu^t}\left[\mathbb{E}_{a_h\sim\mu^t(\cdot|x_h)}\left[\sum_{h'=h}^H r_{h'}\right] - \mathbb{E}_{a_h\sim\mu^\dagger(\cdot|x_h)}\left[\sum_{h'=h}^H r_{h'}\right]\right] \\ &\overset{(ii)}{=} \max_{\mu^\dagger\in\Pi_{\max}} \sum_{t=1}^T \sum_{x_h\in\mathcal{X}_h} \mu_{1:h-1}^\dagger(x_{h-1},a_{h-1}) \cdot \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), L_h^t(x_h,\cdot)\right\rangle\end{aligned}$$

$$= \max_{\mu^\dagger \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu^\dagger_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \mathfrak{R}_h^{\mathrm{imm},T}(x_h).$$

Above, (i) follows as the rewards for the first $h-1$ steps are the same for the two expectations; (ii) follows by definition of the counterfactual loss function (cumulative loss multiplied by the opponent and environment's policy / transition probabilities, as well as the max player's own policy from step h onward). The claim (with the μ form expression) thus follows by renaming the dummy variable $\mu^\dagger$ as μ.

To verify that the second expression is equivalent to the first expression, it suffices to notice that the max over $\mu_{1:h-1} \in \Pi_{\max}$ consists of separable optimization problems over $\mu_{h'}(\cdot|x_{h'})$ over all $x_{h'} \in \mathcal{X}_{h'}$, $h' \le h-1$, due to the perfect recall assumption (different $(x_{h'}, a_{h'})$ leads to disjoint subtrees). Therefore, we can rewrite the above as

$$\mathfrak{R}_h^T = \sum_{x_1 \in \mathcal{X}_1} \max_{\mu_1(\cdot|x_1) \in \Delta(\mathcal{A})} \sum_{a_1 \in \mathcal{A}} \mu_1(a_1|x_1) \sum_{x_2 \in \mathcal{C}(x_1, a_1)} \cdots$$
$$\sum_{x_{h-1} \in \mathcal{C}(x_{h-2}, a_{h-2})} \max_{\mu_{h-1}(\cdot|x_{h-1}) \in \Delta(\mathcal{A})} \sum_{a_{h-1} \in \mathcal{A}} \mu_{h-1}(a_{h-1}|x_{h-1}) \sum_{x_h \in \mathcal{C}(x_{h-1}, a_{h-1})} \mathfrak{R}_h^{\mathrm{imm},T}(x_h).$$

Further noticing (backward recursively) that each max over the action distribution is achieved at a single action yields the claimed sum-max form expression. □

E.4.2 Proof of Theorem 44

We now prove our main theorem on the regret of the CFR algorithm.

By Lemma 135, we have $\widetilde{\mathfrak{R}}^T \le \sum_{h=1}^H \mathfrak{R}_h^T$, where for any $h \in [H]$ we have

$$\mathfrak{R}_h^T = \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \mathfrak{R}_h^{\mathrm{imm},T}(x_h)$$
$$= \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^T \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), L_h^t(x_h, \cdot) \right\rangle$$
$$\le \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \underbrace{\max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^T \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), \widetilde{L}_h^t(x_h, \cdot) \right\rangle}_{:= \widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h)}$$

$$
\begin{aligned}
&+ \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \sum_{t=1}^{T} \left\langle \mu_h^t(\cdot|x_h), L_h^t(x_h, \cdot) - \widetilde{L}_h^t(x_h, \cdot) \right\rangle \\
&+ \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^{T} \left\langle \mu_h^\dagger(\cdot|x_h), \widetilde{L}_h^t(x_h, \cdot) - L_h^t(x_h, \cdot) \right\rangle \\
&\overset{(i)}{=} \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h)}_{:=\mathrm{REGRET}_h} \\
&\quad + \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \sum_{t=1}^{T} \mu_h^t(a_h|x_h) \Big[L_h^t(x_h, a_h) - \widetilde{L}_h^t(x_h, a_h) \Big]}_{:=\mathrm{BIAS}_h^1} \\
&\quad + \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h}(x_h, a_h) \sum_{t=1}^{T} \Big[\widetilde{L}_h^t(x_h, a_h) - L_h^t(x_h, a_h) \Big]}_{:=\mathrm{BIAS}_h^2} \\
&= \mathrm{REGRET}_h + \mathrm{BIAS}_h^1 + \mathrm{BIAS}_h^2.
\end{aligned}
$$

Above, the simplification of the BIAS_h^2 part in (i) uses the fact that the inner max over $\mu_h^\dagger(\cdot|x_h)$ and the outer max over $\mu_{1:(h-1)}$ are separable and thus can be merged into a single max over $\mu_{1:h}$.

We now state three lemmas that bound each term above. Their proofs are deferred to Sections E.4.3-E.4.5.

Lemma 136 (Bound on BIAS_h^1)**.** *For any sequence of opponents' policies $\nu^t \in \mathcal{F}_{t-1}$, using the estimator $\widetilde{L}_h$ in* (6.13)*, with probability $1-\delta/10$, we have*

$$\sum_{h=1}^{H} \mathrm{BIAS}_h^1 \le 2\sqrt{H^3XAT\iota} + HX\iota,$$

where $\iota = \log(10X/\delta)$.

Lemma 137 (Bound on BIAS_h^2)**.** *For any sequence of opponents' policies $\nu^t \in \mathcal{F}_{t-1}$, using the estimator $\widetilde{L}_h$ in* (6.13)*, with probability $1-\delta/10$, we have*

$$\sum_{h=1}^{H} \mathrm{BIAS}_h^2 \le 2\sqrt{H^3XAT\iota} + HXA\iota,$$

where $\iota = \log(10XA/\delta)$.

Lemma 138 (Bound on REGRET_h)**.** *Choosing* $\eta = \sqrt{XA\iota/(H^3T)}$*, we have that with probability at least* $1-\delta/10$ *(over the randomness within the loss estimator* $\widetilde{L}_h^t$*),*

$$\sum_{h=1}^{H} \mathrm{REGRET}_h \le 2\sqrt{H^3XAT\iota} + \sqrt{HX^3A^3\iota^3/(4T)},$$

where $\iota = \log(10XA/\delta)$.

Combining Lemma 136, 137, and 138, we obtain the following: Choosing $\eta = \sqrt{XA\iota/(H^3T)}$, with probability at least $1-3\delta/10 \ge 1-\delta$, we have

$$\begin{aligned}\widetilde{\mathfrak{R}}^T &\le \sum_{h=1}^{H}\mathfrak{R}_h^T \le \sum_{h=1}^{H}\mathrm{REGRET}_h + \sum_{h=1}^{H}\mathrm{BIAS}_h^1 + \sum_{h=1}^{H}\mathrm{BIAS}_h^2 \\ &\le 6\sqrt{H^3XAT\iota} + 2HXA\iota + \sqrt{HX^3A^3\iota^3/(4T)}.\end{aligned}$$

Additionally, recall the naive bound $\widetilde{\mathfrak{R}}^T \le HT$ on the regret (which follows as $\langle \mu^t, \ell^t\rangle \in [0,H]$ for any $\mu \in \Pi_{\max}$, $t\in[T]$), we get

$$\begin{aligned}\widetilde{\mathfrak{R}}^T &\le \min\left\{6\sqrt{H^3XAT\iota} + 2HXA\iota + \sqrt{HX^3A^3\iota^3/4T}, HT\right\} \\ &\le HT\cdot\min\left\{6\sqrt{HXA\iota/T} + 2XA\iota/T + \sqrt{X^3A^3\iota^3/(4HT^3)}, 1\right\}.\end{aligned}$$

For $T > HXA\iota$, the min above is upper bounded by $9\sqrt{HXA\iota/T}$. For $T \le HXA\iota$, the min above is upper bounded by $1 \le 9\sqrt{HXA\iota/T}$. Therefore, we always have

$$\widetilde{\mathfrak{R}}^T \le HT\cdot 9\sqrt{HXA\iota/T} = 9\sqrt{H^3XAT\iota}.$$

This is the desired result. □

E.4.3 Proof of Lemma 136

Rewrite BIAS_h^1 as

$$\begin{aligned}\mathrm{BIAS}_h^1 = \max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} &\frac{\mu_{1:(h-1)}(x_{h-1},a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})} \\ &\cdot \sum_{t=1}^{T} \mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})\mu_h^t(a_h|x_h)\cdot\Big[L_h^t(x_h,a_h) - \widetilde{L}_h^t(x_h,a_h)\Big] \\ = \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} &\frac{\mu_{1:(h-1)}(x_{h-1},a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})} \cdot \sum_{t=1}^{T} \widetilde{\Delta}_t^{x_h},\end{aligned}$$

where the random variable $\widetilde{\Delta}_t^{x_h}$ is defined by

$$\sum_{a_h\in\mathcal{A}} \frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\left[\mu^{\star,h}_{1:h}(x_h,a_h)L_h^t(x_h,a_h) - \left(H-h+1-\sum_{h'=h}^{H} r_{h'}^{t,(h)}\right)\mathbf{1}\left\{(x_h^{t,(h)},a_h^{t,(h)}) = (x_h,a_h)\right\}\right] \tag{E.8}$$

Observe that the random variables $\widetilde{\Delta}_t^{x_h}$ satisfy the following:

- $\widetilde{\Delta}_t^{x_h} \le H$ almost surely:

$$\begin{aligned}\widetilde{\Delta}_t^{x_h} &\le \sum_{a_h\in\mathcal{A}} \frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\cdot\mu^{\star,h}_{1:h}(x_h,a_h)L_h^t(x_h,a_h) \\ &= \sum_{a_h\in\mathcal{A}} \mu_h^t(a_h|x_h)\mu^{\star,h}_{1:(h-1)}(x_{h-1},a_{h-1})L_h^t(x_h,a_h) \le H.\end{aligned}$$

  Above, the last bound follows from Lemma 110(a).

- $\mathbb{E}[\widetilde{\Delta}_t^{x_h}|\mathcal{F}_{t-1}] = 0$, where $\mathcal{F}_{t-1}$ is the σ-algebra containing all information after iteration $t-1$;

- The conditional variance $\mathbb{E}[(\widetilde{\Delta}_t^{x_h})^2|\mathcal{F}_{t-1}]$ can be bounded as

$$\begin{aligned}&\mathbb{E}\left[\left(\widetilde{\Delta}_t^{x_h}\right)^2\Big|\mathcal{F}_{t-1}\right] \\ &\le \mathbb{E}\left[\sum_{a_h\in\mathcal{A}}\left(\frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\right)^2\cdot\left(H-h+1-\sum_{h'=h}^{H} r_{h'}^{t,(h)}\right)^2\mathbf{1}\left\{(x_h^{t,(h)},a_h^{t,(h)}) = (x_h,a_h)\right\}\Big|\mathcal{F}_{t-1}\right]\end{aligned}$$

$$
\begin{aligned}
&\le H^2 \sum_{a_h\in\mathcal{A}} \left(\frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\right)^2 \cdot \mathbb{P}^{\mu^{\star,h}_{1:h},\nu^t}\Big((x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\Big) \\
&= H^2 \sum_{a_h\in\mathcal{A}} \left(\frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\right)^2 \cdot \mu^{\star,h}_{1:h}(x_h, a_h) \cdot p^{\nu^t}_{1:h}(x_h) \\
&= H^2 \sum_{a_h\in\mathcal{A}} \underbrace{\left(\frac{\mu_h^t(a_h|x_h)}{\mu_h^{\star,h}(a_h|x_h)}\right)}_{\le A} \cdot \mu^{\star,h}_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \mu_h^t(a_h|x_h) p^{\nu^t}_{1:h}(x_h) \\
&\le H^2 A \cdot \sum_{a_h\in\mathcal{A}} \mu^{\star,h}_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \mu_h^t(a_h|x_h) p^{\nu^t}_{1:h}(x_h).
\end{aligned}
$$

Therefore, we can apply Freedman's inequality (Lemma 119) and union bound to get that, for any fixed $\lambda \in (0, 1/H]$, with probability at least $1-\delta/10$, the following holds simultaneously for all (h, x_h):

$$
\sum_{t=1}^{T} \widetilde{\Delta}_t^{x_h} \le \lambda H^2 A \sum_{a_h\in\mathcal{A}} \mu^{\star,h}_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \sum_{t=1}^{T} \mu_h^t(a_h|x_h) p^{\nu^t}_{1:h}(x_h) + \frac{\iota}{\lambda},
$$

where $\iota := \log(10X/\delta)$ is a log factor. Plugging this bound into (E.8) yields that, for all $h \in [H]$,

$$
\begin{aligned}
&\mathrm{BIAS}_h^1 = \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1}, a_{h-1})} \cdot \sum_{t=1}^{T} \widetilde{\Delta}_t^{x_h} \\
&\le \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1}, a_{h-1})} \cdot \left[\lambda H^2 A \sum_{a_h\in\mathcal{A}} \mu^{\star,h}_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \sum_{t=1}^{T} \mu_h^t(a_h|x_h) p^{\nu^t}_{1:h}(x_h) + \frac{\iota}{\lambda}\right] \\
&\le \lambda H^2 A \cdot \max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h-1}(x_{h-1}, a_{h-1}) \sum_{t=1}^{T} \mu_h^t(a_h|x_h) p^{\nu^t}_{1:h}(x_h) \\
&\qquad + \frac{\iota}{\lambda} \cdot \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu^{\star,h}_{1:(h-1)}(x_{h-1}, a_{h-1})} \\
&\overset{(i)}{=} \lambda H^2 AT + \frac{\iota}{\lambda} \cdot \frac{1}{A} \max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \frac{(\mu_{1:(h-1)}\mu_h^{\mathrm{unif}})(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \\
&\overset{(ii)}{=} \lambda H^2 AT + \frac{\iota}{\lambda} \cdot X_h.
\end{aligned}
$$

Above, (i) used the fact that $\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\mu_{1:h-1}(x_{h-1},a_{h-1})\mu_h^t(a_h|x_h)p_{1:h}^{\nu^t}(x_h)=1$ for any $\mu\in\Pi_{\max}$ and any $t\in[T]$ (Lemma 110(a)), as well as the fact that $\mu_h^{\star,h}(a_h|x_h)=\mu_h^{\mathrm{unif}}(a_h|x_h):=1/A$; (ii) used the balancing property of $\mu_{1:h}^{\star,h}$ (Lemma 37). Combining the bounds for all $h\in[H]$, we get that with probability at least $1-\delta/10$,

$$\sum_{h=1}^{H}\mathrm{BIAS}_h^1\le\lambda H^3AT+\frac{X\iota}{\lambda}.$$

Choosing

$$\lambda=\min\left\{\sqrt{\frac{X\iota}{H^3AT}},\frac{1}{H}\right\}\le\frac{1}{H},$$

we obtain the bound

$$\sum_{h=1}^{H}\mathrm{BIAS}_h^1\le2\sqrt{H^3XAT\iota}+HX\iota.$$

This is the desired result. □

E.4.4 Proof of Lemma 137

The proof strategy is similar to Lemma 136. We can rewrite BIAS_h^2 as

$$\begin{aligned}\mathrm{BIAS}_h^2&=\max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\frac{\mu_{1:h}(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)}\cdot\sum_{t=1}^{T}\mu_{1:h}^{\star,h}(x_h,a_h)\Big[\widetilde{L}_h^t(x_h,a_h)-L_h^t(x_h,a_h)\Big]\\&=\max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\frac{\mu_{1:h}(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)}\cdot\sum_{t=1}^{T}\Delta_t^{x_h,a_h},\end{aligned}$$

where the last equality used the definition of the loss estimator $\widetilde{L}_h^t(x_h,a_h)$ in (6.13) and the random variable $\Delta_t^{x_h,a_h}$ is defined by

$$\left[\left(H-h+1-\sum_{h'=h}^{H}r_{h'}^{t,(h)}\right)\mathbf{1}\left\{(x_h^{t,(h)},a_h^{t,(h)})=(x_h,a_h)\right\}-\mu_{1:h}^{\star,h}(x_h,a_h)L_h^t(x_h,a_h)\right]\tag{E.9}$$

.

Observe that the random variables $\Delta_t^{x_h,a_h}$ satisfy the following:

- $\Delta_t^{x_h,a_h} \le H$ almost surely.
- $\mathbb{E}[\Delta_t^{(x_h,a_h)}|\mathcal{F}_{t-1}] = 0$, where $\mathcal{F}_{t-1}$ is the σ-algebra containing all information after iteration $t-1$. This follows as the episode was sampled using $\mu^{t,(h)} = \mu_{1:h}^{\star,h}\mu_{h+1:H}^{t}$, as well as the definition of $L_h^t(x_h,a_h)$ in (6.11).
- The conditional variance $\mathbb{E}[(\Delta_t^{(x_h,a_h)})^2|\mathcal{F}_{t-1}]$ can be bounded as

$$\begin{aligned}
&\mathbb{E}\left[\left(\Delta_t^{(x_h,a_h)}\right)^2\middle|\mathcal{F}_{t-1}\right] \\
\le&\mathbb{E}\left[\left(H-h+1-\sum_{h'=h}^{H} r_{h'}^{t,(h)}\right)^2 \mathbf{1}\left\{(x_h^{t,(h)},a_h^{t,(h)})=(x_h,a_h)\right\}\middle|\mathcal{F}_{t-1}\right] \\
&\le H^2\mathbb{P}^{\mu_{1:h}^{\star,h},\nu^t}\left((x_h^{t,(h)},a_h^{t,(h)})=(x_h,a_h)\right) \\
&= H^2\mu_{1:h}^{\star,h}(x_h,a_h)\cdot p_{1:h}^{\nu^t}(x_h).
\end{aligned}$$

Therefore, we can apply Freedman's inequality (Lemma 119) and union bound to get that, for any fixed $\lambda \in (0,1/H]$, with probability at least $1-\delta/10$, the following holds simultaneously for all (h,x_h,a_h):

$$\sum_{t=1}^{T}\Delta_t^{(x_h,a_h)} \le \lambda H^2\mu_{1:h}^{\star,h}(x_h,a_h)\cdot\sum_{t=1}^{T}p_{1:h}^{\nu^t}(x_h)+\frac{\iota}{\lambda},$$

where $\iota := \log(10XA/\delta)$ is a log factor. Plugging this bound into (E.9) yields that, for all $h\in[H]$,

$$\begin{aligned}
&\mathrm{BIAS}_h^2 = \max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\frac{\mu_{1:h}(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)}\cdot\sum_{t=1}^{T}\Delta_t^{x_h,a_h} \\
&\le \max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\frac{\mu_{1:h}(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)}\cdot\left[\lambda H^2\mu_{1:h}^{\star,h}(x_h,a_h)\cdot\sum_{t=1}^{T}p_{1:h}^{\nu^t}(x_h)+\frac{\iota}{\lambda}\right] \\
&\le \lambda H^2\cdot\max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\mu_{1:h}(x_h,a_h)\sum_{t=1}^{T}p_{1:h}^{\nu^t}(x_h)+\frac{\iota}{\lambda}\cdot\max_{\mu\in\Pi_{\max}}\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}\frac{\mu_{1:h}(x_h,a_h)}{\mu_{1:h}^{\star,h}(x_h,a_h)}
\end{aligned}$$

$$\overset{(i)}{=} \lambda H^2 T + \frac{\iota}{\lambda} \cdot X_h A.$$

Above, (i) used the fact that $\sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h}(x_h, a_h) p^{\nu^t}_{1:h}(x_h) = 1$ for any $\mu \in \Pi_{\max}$ and any $t \in [T]$ (Lemma 110(a)), as well as the balancing property of $\mu^{\star,h}_{1:h}$ (Lemma 37). Combining the bounds for all $h \in [H]$, we get that with probability at least $1 - \delta/10$,

$$\sum_{h=1}^{H} \mathrm{BIAS}_h^2 \le \lambda H^3 T + \frac{XA\iota}{\lambda}.$$

Choosing

$$\lambda = \min\left\{\sqrt{\frac{XA\iota}{H^3 T}}, \frac{1}{H}\right\} \le \frac{1}{H},$$

we obtain the bound

$$\sum_{h=1}^{H} \mathrm{BIAS}_h^2 \le 2\sqrt{H^3 XAT\iota} + HXA\iota.$$

This is the desired result. □

E.4.5 Proof of Lemma 138

Recall that for all (h, x_h), we have implemented Line 8 of Algorithm 11 as the Hedge algorithm (Algorithm 21) with learning rate $\eta\mu^{\star,h}_{1:h}(x_h, a)$ and loss vector $\left\{\widetilde{L}^t_h(x_h, a)\right\}_{a\in\mathcal{A}}$ (cf. (6.12)). Therefore, applying Lemma 115, the standard regret bound for HEDGE, we get for arbitrary $a \in \mathcal{A}$

$$
\begin{aligned}
&\widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h) = \max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^T \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), \widetilde{L}_h^t(x_h, \cdot) \right\rangle \\
&\le \frac{\log A}{\eta \mu_{1:h}^{\star,h}(x_h, a)} + \frac{\eta}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \cdot \mu_h^t(a_h|x_h) \Big(\widetilde{L}_h^t(x_h, a_h) \Big)^2 \\
&\overset{(i)}{=} \frac{\log A}{\eta \mu_{1:h}^{\star,h}(x_h, a)} \\
&+ \frac{\eta}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \mu_h^t(a_h|x_h) \cdot \frac{\Big(H - h + 1 - \sum_{h'=h}^H r_{h'}^{t,(h)} \Big)^2 \mathbf{1}\Big\{ (x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h) \Big\}}{\Big(\mu_{1:h}^{\star,h}(x_h, a_h) \Big)^2} \\
&\le \frac{\log A}{\eta \mu_{1:h}^{\star,h}(x_h, a)} + \frac{\eta H^2}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} \mu_h^t(a_h|x_h) \cdot \frac{\mathbf{1}\Big\{ (x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h) \Big\}}{\mu_{1:h}^{\star,h}(x_h, a_h)}.
\end{aligned}
\tag{E.10}
$$

Above, (i) used the form of $\widetilde{L}_h^t$ in (6.13). Plugging this into the definition of REGRET_h, we have

$$
\begin{aligned}
&\mathrm{REGRET}_h = \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h) \\
&\le \underbrace{\max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \frac{\log A}{\eta \mu_{1:h}^{\star,h}(x_h, a)}}_{\mathrm{I}_h} \\
&+ \underbrace{\max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \frac{\eta H^2}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} \mu_h^t(a_h|x_h) \cdot \frac{\mathbf{1}\Big\{ (x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h) \Big\}}{\mu_{1:h}^{\star,h}(x_h, a_h)}}_{\mathrm{II}_h}.
\end{aligned}
\tag{E.11}
$$

We first calculate term I_h. We have

$$
\begin{aligned}
\mathrm{I}_h &\overset{(i)}{=} \frac{\log A}{\eta} \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \frac{1}{A_h} \cdot \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu_{1:h}^{\star,h}(x_h, a_h)} \\
&= \frac{\log A}{\eta} \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \frac{(\mu_{1:(h-1)} \mu_h^{\mathrm{unif}})(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \\
&\overset{(ii)}{=} \frac{\log A}{\eta} \cdot X_h A = \frac{X_h A \log A}{\eta},
\end{aligned}
$$

where (i) follows by splitting the sum over a_h and using the fact that $\mu^{\star,h}_{1:h}(x_h, a)$ does not depend on a; (ii) follows from the balancing property of $\mu^{\star,h}_{1:h}$ (Lemma 37).

Next, we bound term II_h. We have

$$\begin{aligned}
&\mathrm{II}_h \\
=&\frac{\eta H^2}{2} \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \sum_{t=1}^{T} \sum_{a_h\in\mathcal{A}} \mu^t_h(a_h|x_h) \cdot \frac{\mathbf{1}\left\{(x^{t,(h)}_h, a^{t,(h)}_h) = (x_h, a_h)\right\}}{\mu^{\star,h}_{1:h}(x_h, a_h)} \\
=&\frac{\eta H^2}{2} \max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu^{\star,h}_{1:h}(x_h, a)} \cdot \sum_{t=1}^{T} \underbrace{\sum_{a_h\in\mathcal{A}} \mu^t_h(a_h|x_h) \cdot \mathbf{1}\left\{(x^{t,(h)}_h, a^{t,(h)}_h) = (x_h, a_h)\right\}}_{:=\overline{\Delta}^{x_h}_t}.
\end{aligned} \tag{E.12}$$

The last equality above used the fact that $\mu^{\star,h}_{1:h}(x_h, a_h)$ does not depend on a_h (cf. (6.1)).

Observe that the random variables $\overline{\Delta}^{x_h}_t$ satisfy the following:

- $\overline{\Delta}^{x_h}_t \in [0, 1]$ almost surely;
- $\mathbb{E}[\overline{\Delta}^{x_h}_t|\mathcal{F}_{t-1}] = \sum_{a_h\in\mathcal{A}} \mu^{\star,h}_{1:h}(x_h, a_h)\cdot\mu^t_h(a_h|x_h)p^{\nu^t}_{1:h}(x_h)$, where $\mathcal{F}_{t-1}$ is the σ-algebra containing all information after iteration $t-1$;
- The conditional variance $\mathrm{Var}[\overline{\Delta}^{x_h}_t|\mathcal{F}_{t-1}]$ can be bounded as

$$\begin{aligned}
&\mathrm{Var}\left[\overline{\Delta}^{x_h}_t\middle|\mathcal{F}_{t-1}\right] \le \mathbb{E}\left[\left(\overline{\Delta}^{x_h}_t\right)^2\middle|\mathcal{F}_{t-1}\right] \\
=&\,\mathbb{E}\left[\sum_{a_h\in\mathcal{A}} \left(\mu^t_h(a_h|x_h)\right)^2 \mathbf{1}\left\{(x^{t,(h)}_h, a^{t,(h)}_h) = (x_h, a_h)\right\}\middle|\mathcal{F}_{t-1}\right] \\
=&\sum_{a_h\in\mathcal{A}} \left(\mu^t_h(a_h|x_h)\right)^2 \cdot \mathbb{P}^{\mu^{\star,h}_{1:h}\times\nu^t}\left((x^{t,(h)}_h, a^{t,(h)}_h) = (x_h, a_h)\right) \\
=&\sum_{a_h\in\mathcal{A}} \mu^{\star,h}_{1:h}(x_h, a_h) \cdot \left(\mu^t_h(a_h|x_h)\right)^2 \cdot p^{\nu^t}_{1:h}(x_h).
\end{aligned}$$

Therefore, we can apply Freedman's inequality (Lemma 119) and a union bound to obtain that, for any $\lambda \in (0, 1]$, with probability at least $1-\delta/10$, the following holds simultaneously for all (h, x_h):

$$\begin{aligned}
&\sum_{t=1}^{T} \overline{\Delta}_t^{x_h} - \sum_{t=1}^{T} \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \cdot \mu_h^t(a_h|x_h) p_{1:h}^{\nu^t}(x_h) \\
&\le \lambda \cdot \sum_{t=1}^{T} \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \cdot \big(\mu_h^t(a_h|x_h)\big)^2 \cdot p_{1:h}^{\nu^t}(x_h) + \frac{\iota}{\lambda},
\end{aligned}$$

where $\iota := \log(10X/\delta)$ is a log factor. Plugging this bound into (E.12) yields that, for all $h \in [H]$,

$$\begin{aligned}
\mathrm{II}_h &\le \frac{\eta H^2}{2} \cdot \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu_{1:h}^{\star,h}(x_h, a)} \cdot \sum_{t=1}^{T} \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \cdot \mu_h^t(a_h|x_h) p_{1:h}^{\nu^t}(x_h) \\
&\quad + \frac{\eta H^2}{2} \cdot \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu_{1:h}^{\star,h}(x_h, a)} \\
&\qquad\qquad \cdot \left[\lambda \sum_{t=1}^{T} \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h, a_h) \cdot \big(\mu_h^t(a_h|x_h)\big)^2 \cdot p_{1:h}^{\nu^t}(x_h) + \frac{\iota}{\lambda}\right] \\
&\overset{(i)}{\le} \frac{\eta H^2}{2} \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \sum_{t=1}^{T} \mu_h^t(a_h|x_h) p_{1:h}^{\nu^t}(x_h) \\
&\quad + \frac{\eta H^2}{2} \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \lambda \sum_{t=1}^{T} \big(\mu_h^t(a_h|x_h)\big)^2 \cdot p_{1:h}^{\nu^t}(x_h) \\
&\quad + \frac{\eta H^2}{2} \cdot \frac{\iota}{\lambda} \cdot \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \frac{\mu_{1:(h-1)}(x_{h-1}, a_{h-1})}{\mu_{1:h}^{\star,h}(x_h, a)} \\
&\overset{(ii)}{\le} \frac{\eta H^2}{2}(1 + \lambda) \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \sum_{t=1}^{T} \mu_h^t(a_h|x_h) p_{1:h}^{\nu^t}(x_h) \\
&\quad + \frac{\eta H^2}{2} \cdot \frac{\iota}{\lambda} \cdot \max_{\mu \in \Pi_{\max}} \sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \frac{(\mu_{1:(h-1)} \mu_h^{\mathrm{unif}})(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \\
&\overset{(iii)}{=} \frac{\eta H^2}{2}(1 + \lambda) T + \frac{\eta H^2}{2} \cdot \frac{\iota}{\lambda} \cdot X_h A.
\end{aligned}$$

Above, (i) used again the fact that $\mu_{1:h}^{\star,h}(x_h, a) = \mu_{1:h}^{\star,h}(x_h, a_h)$ for any $a, a_h \in \mathcal{A}$; (ii) used the fact that $\mu_h^t(a_h|x_h) \le 1$; (iii) used the fact that

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} (\mu_{1:(h-1)} \mu_h^t)(x_h, a_h) p_{1:h}^{\nu^t}(x_h) = 1$$

for any $\mu \in \Pi_{\max}$ and any $t \in [T]$ (Lemma 110(a)), as well as the balancing property of $\mu_{1:h}^{\star,h}$ (Lemma 37).

Combining the bounds for I_h and II_h, we obtain that

$$\begin{aligned}
\sum_{h=1}^{H} \mathrm{REGRET}_h &\le \sum_{h=1}^{H} (\mathrm{I}_h + \mathrm{II}_h) \\
&\le \sum_{h=1}^{H} \left[\frac{X_h A \log A}{\eta} + \frac{\eta H^2}{2}(1+\lambda)T + \frac{\eta H^2 X_h A \iota}{2\lambda} \right] \\
&\le \frac{XA\iota}{\eta} + \frac{\eta H^3}{2} T + \frac{\eta H^2}{2} \left[\lambda \cdot HT + \frac{XA\iota}{\lambda} \right],
\end{aligned}$$

where we have redefined the log factor $\iota := \log(10XA/\delta)$. Choosing $\lambda = 1$, the above can be upper bounded by

$$\frac{XA\iota}{\eta} + \eta H^3 T + \frac{\eta H^2 XA\iota}{2}.$$

Further choosing $\eta = \sqrt{XA\iota/(H^3 T)}$, we obtain the bound

$$\sum_{h=1}^{H} \mathrm{REGRET}_h \le 2\sqrt{H^3 XAT\iota} + \sqrt{HX^3A^3\iota^3/(4T)}.$$

This is the desired result. □

E.4.6 Proof of Theorem 46

The proof is similar as Theorem 44, except for plugging in the regret bound for Regret Matching instead of Hedge.

First, by Lemma 135, we have $\widetilde{\mathfrak{R}}^T \le \sum_{h=1}^H \mathfrak{R}_h^T$, where for any $h \in [H]$ we have

$$
\begin{aligned}
\mathfrak{R}_h^T \le\ & \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{x_h\in\mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h)}_{:=\mathrm{REGRET}_h} \\
& + \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \sum_{t=1}^T \mu_h^t(a_h|x_h) \Big[L_h^t(x_h, a_h) - \widetilde{L}_h^t(x_h, a_h)\Big]}_{:=\mathrm{BIAS}_h^1} \\
& + \underbrace{\max_{\mu\in\Pi_{\max}} \sum_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}} \mu_{1:h}(x_h, a_h) \sum_{t=1}^T \Big[\widetilde{L}_h^t(x_h, a_h) - L_h^t(x_h, a_h)\Big]}_{:=\mathrm{BIAS}_h^2} \\
=\ & \mathrm{REGRET}_h + \mathrm{BIAS}_h^1 + \mathrm{BIAS}_h^2,
\end{aligned}
\tag{E.13}
$$

where the definition of $\widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h)$, $L_h^t(x_h, a_h)$ are at the beginning of Section E.4.1 and the definition of $\widetilde{L}_h^t(x_h, a_h)$ are given by Algorithm 11.

To upper bound BIAS_h^1 and BIAS_h^2, we use the same strategy as the proof of Lemma 136 and 137 (whose proofs are independent of the regret minimizer), so that we have the same bound as in Lemma 136 and 137: with probability at least $1-\delta/5$, we have

$$
\sum_{h=1}^H \mathrm{BIAS}_h^1 \le 2\sqrt{H^3XAT\iota} + HX\iota, \qquad \sum_{h=1}^H \mathrm{BIAS}_h^2 \le 2\sqrt{H^3XAT\iota} + HXA\iota, \tag{E.14}
$$

where $\iota = \log(10XA/\delta)$.

To upper bound REGRET_h, we use the same strategy as the proof of Lemma 138 as in Section E.4.5. First, applying the regret bound for Regret Matching (Lemma 116

& Remark 117), we get (below $a \in \mathcal{A}$ is arbitrary, and $\eta > 0$ is also arbitrary)

$$\begin{aligned}
&\widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h) = \max_{\mu_h^\dagger(\cdot|x_h)} \sum_{t=1}^T \left\langle \mu_h^t(\cdot|x_h) - \mu_h^\dagger(\cdot|x_h), \widetilde{L}_h^t(x_h,\cdot) \right\rangle \\
\le{}& \frac{1}{\eta \mu_{1:h}^{\star,h}(x_h,a)} + \frac{\eta}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} \mu_{1:h}^{\star,h}(x_h,a_h) \cdot A \overline{\mu}_h^t(a_h|x_h) \Big(\widetilde{L}_h^t(x_h,a_h) \Big)^2 \qquad \text{(E.15)} \\
\le{}& \frac{1}{\eta \mu_{1:h}^{\star,h}(x_h,a)} + \frac{\eta H^2}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} A \cdot \overline{\mu}_h^t(a_h|x_h) \cdot \frac{\mathbf{1}\left\{ (x_h^{t,(h)}, a_h^{t,(h)}) = (x_h,a_h) \right\}}{\mu_{1:h}^{\star,h}(x_h,a_h)},
\end{aligned}$$

where $\overline{\mu}_h^t(a_h|x_h) = (\mu_h^t(a_h|x_h) + (1/A))/2$ is a probability distribution over $[A]$. Comparing the right hand side of Eq. (E.15) with the right hand side of Eq. (E.10), we can see that there is only one difference which is $A \cdot \overline{\mu}_h^t$ versus μ_h^t. Plugging this into the definition of REGRET_h, we have

$$\begin{aligned}
&\mathrm{REGRET}_h = \max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \widetilde{\mathfrak{R}}_h^{\mathrm{imm},T}(x_h) \\
\le{}& \underbrace{\max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \frac{1}{\eta \mu_{1:h}^{\star,h}(x_h,a)}}_{\mathrm{I}_h} \\
+{}& \underbrace{\max_{\mu \in \Pi_{\max}} \sum_{x_h \in \mathcal{X}_h} \mu_{1:(h-1)}(x_{h-1}, a_{h-1}) \cdot \frac{\eta H^2}{2} \cdot \sum_{t=1}^T \sum_{a_h \in \mathcal{A}} A \cdot \overline{\mu}_h^t(a_h|x_h) \cdot \frac{\mathbf{1}\left\{ (x_h^{t,(h)}, a_h^{t,(h)}) = (x_h,a_h) \right\}}{\mu_{1:h}^{\star,h}(x_h,a_h)}}_{\mathrm{II}_h}.
\end{aligned} \qquad \text{(E.16)}$$

Comparing Eq. (E.16) with Eq. (E.11), we can see that I_h in Eq. (E.16) is the same as I_h in Eq. (E.11), and II_h in Eq. (E.16) and (E.11) only have one difference which is also $A \cdot \overline{\mu}_h^t$ versus μ_h^t. Using the same argument as in the former proof, we have

$$\mathrm{I}_h = \frac{X_h A}{\eta}.$$

Furthermore, using the same argument as in the former proof, we can show that the upper bound of II_h in Eq. (E.16) is at most A times the upper bound of II_h in Eq.

(E.11). This gives for any $\lambda \in (0,1)$, with probability at least $1-\delta/10$, we have

$$\mathrm{II}_h \le \frac{\eta H^2 A}{2}(1+\lambda)T + \frac{\eta H^2}{2} \cdot \frac{\iota}{\lambda} \cdot X_h A^2.$$

Combining the bounds for I_h and II_h, we obtain that

$$\sum_{h=1}^{H} \mathrm{REGRET}_h \le \sum_{h=1}^{H} (\mathrm{I}_h + \mathrm{II}_h) \le \frac{XA}{\eta} + \frac{\eta H^3 A}{2} T + \frac{\eta H^2 A}{2}\left[\lambda \cdot HT + \frac{XA\iota}{\lambda}\right],$$

Choosing $\lambda = 1$ and choosing $\eta = \sqrt{X\iota/(H^3 T)}$, with probability at least $1-\delta/10$, we obtain the bound

$$\sum_{h=1}^{H} \mathrm{REGRET}_h \le 2\sqrt{H^3 X A^2 T \iota} + \sqrt{H X^3 A^4 \iota^3/(4T)}. \tag{E.17}$$

This bound is $\sqrt{A}$ times larger than the bound of $\sum_{h=1}^{H} \mathrm{REGRET}_h$ as in Lemma 138.

Combining Eq. (E.13), (E.14) and (E.17), we obtain the following: with probability at least $1 - 3\delta/10 \ge 1-\delta$, we have

$$\begin{aligned}\widetilde{\mathfrak{R}}^T &\le \sum_{h=1}^{H} \mathfrak{R}_h^T \le \sum_{h=1}^{H} \mathrm{REGRET}_h + \sum_{h=1}^{H} \mathrm{BIAS}_h^1 + \sum_{h=1}^{H} \mathrm{BIAS}_h^2 \\ &\le 6\sqrt{H^3 X A^2 T \iota} + 2HXA\iota + \sqrt{H X^3 A^4 \iota^3/(4T)}.\end{aligned}$$

Additionally, recall the naive bound $\widetilde{\mathfrak{R}}^T \le HT$ on the regret (which follows as $\langle \mu^t, \ell^t \rangle \in [0, H]$ for any $\mu \in \Pi_{\max}$, $t \in [T]$), we get

$$\begin{aligned}\widetilde{\mathfrak{R}}^T &\le \min\left\{6\sqrt{H^3 X A^2 T \iota} + 2HXA\iota + \sqrt{H X^3 A^4 \iota^3/4T}, HT\right\} \\ &\le HT \cdot \min\left\{6\sqrt{HXA^2\iota/T} + 2XA\iota/T + \sqrt{X^3 A^4 \iota^3/(4HT^3)}, 1\right\}.\end{aligned}$$

For $T > HXA^2\iota$, the min above is upper bounded by $9\sqrt{HXA^2\iota/T}$. For $T \le HXA^2\iota$, the min above is upper bounded by $1 \le 9\sqrt{HXA^2\iota/T}$. Therefore, we always have

$$\widetilde{\mathfrak{R}}^T \le HT \cdot 9\sqrt{HXA^2\iota/T} = 9\sqrt{H^3 X A^2 T \iota}.$$

This is the desired result. □

E.5 Proofs for Section 6.4

Regret and CCE Similar as how regret minimization in two-player zero-sum games leads to an approximate Nash equilibrium (Proposition 31), in multi-player general-sum games, regret minimization is known to lead to an approximate NFCCE. Let $\{\pi^t\}_{t=1}^T$ denote a sequence of joint policies (for all players) over T rounds. The regret of the i-th player is defined by

$$\mathfrak{R}_i^T := \max_{\pi_i^\dagger \in \Pi_i} \sum_{t=1}^{T} \left(V_i^{\pi_i^\dagger, \pi_{-i}^t} - V_i^{\pi^t} \right).$$

where Π_i denotes the set of all possible policies for the i-th player.

Using online-to-batch conversion, it is a standard result that sub-linear regret for all the players ensures that the average policy $\overline{\pi}$ is an approximate NFCCE [Celli et al., 2019a].

Proposition 139 (Regret-to-CCE conversion for multi-player general-sum games)**.** *Let the average policy $\overline{\pi}$ be defined as playing a policy within $\{\pi^t\}_{t=1}^T$ uniformly at random, then we have*

$$\mathrm{CCEGap}(\overline{\pi}) = \frac{\max_{i\in[m]} \mathfrak{R}_i^T}{T}.$$

We include a short justification for this standard result here for completeness.

Proof. By definition of $\overline{\pi}$, we have for any $i \in [m]$ and $\pi_i^\dagger \in \Pi_i$ that

$$V_i^{\pi_i^\dagger, \overline{\pi}_{-i}} - V_i^{\overline{\pi}} = \frac{1}{T} \sum_{t=1}^{T} \left(V_i^{\pi_i^\dagger, \pi_{-i}^t} - V_i^{\pi^t} \right).$$

Taking the max over $\pi_i^\dagger \in \Pi_i$ and $i \in [m]$ on both sides yields the desired result. □

E.5.1 Proof of Theorem 49

It is straightforward to see that the regret guarantees for Balanced OMD (Theorem 42) and Balanced CFR (Theorem 44) also hold in multi-player general-sum games (e.g. by modeling all other players as a single opponent). Therefore, the regret-to-CCE conversion in Proposition 139 directly implies that, letting $\overline{\pi}$ denote the joint policy of playing a uniformly sampled policy within $\{\pi^t\}_{t=1}^T$, we have for Balanced OMD that

$$\mathrm{CCEGap}(\overline{\pi}) \le \mathcal{O}\left(\frac{\max_{i\in[m]}\sqrt{H^3X_iA_i\iota T}}{T}\right) = \mathcal{O}\left(\sqrt{H^3\left(\max_{i\in[m]} X_iA_i\right)\iota/T}\right),$$

with probability at least $1-\delta$, where $\iota := \log(3H\sum_{i=1}^m X_iA_i/\delta)$ is a log factor. Choosing $T \ge \widetilde{\mathcal{O}}\big(H^3\big(\max_{i\in[m]} X_iA_i\big)\iota/\varepsilon^2\big)$ ensures that the right-hand side is at most ε. This shows part (a). A similar argument can be done for the Balanced CFR algorithm to show part (b). □

Appendix F

Proofs for Chapter 7

F.1 Proofs for Section 7.1.1 & 7.1.2

F.1.1 Incremental (OMD) form of Algorithm 13

We first present an incremental update of (λ^{t+1}, m^{t+1}) from (λ^t, m^t) as in Algorithm 24. We set the initial values of these variables as

$$\lambda^1_{x_g a_g} \propto_{x_g a_g} \exp\Big\{F^0_{x_g}\Big\}, \qquad m^1_{x_g a_g, h}(a_h|x_h) \propto_{a_h} \exp\Big\{\sum_{x_{h+1}\in\mathcal{C}(x_h, a_h)} F^0_{x_{h+1}}\Big\}, \tag{F.1}$$

where for any $x_h \succeq x_g$, $F^0_{x_h}$ is recursively defined as

$$F^0_{x_h} := \log \sum_{a_h \in \mathcal{A}} \exp\Big\{\sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F^0_{x_{h+1}}\Big\}.$$

Here $F^0_{x_h}$ has an intuitive meaning: it is the logarithm of the number of deterministic sequence-form policies starting from x_h, and can be computed by the above sum-product formulation.

Algorithm 24 is computationally more efficient than Algorithm 13 when the loss estimator is sparse. For example, with bandit feedback, we need to update the loss matrix for at most H infoset-action pairs, and thus incur at most H^3 operations in Algorithm 24. On the contrary, Algorithm 13 requires $O((XA)^2)$ operations to

Algorithm 24 EFCE-OMD (OMD form; equivalent FTRL form in Algorithm 13)

```
Require: Learning rate $\eta$.
Initialize $\lambda^1_{x_ga_g}$, and $m^1_{x_ga_g,h}(a_h|x_h)$, for all $(g, x_g, a_g, h, x_h, a_h)$ with $g \leq h$ using Eq. (F.1).
for $t = 1, 2, \dots, T$ do
    Compute $\phi^t = \phi(\lambda^t, m^t)$ where $\phi$ is in Eq. (7.7).
    Compute the policy $\mu^t$, which is a solution of the fixed point equation $\mu = \phi^t\mu$.
    Receive loss $\ell^t = \{\ell^t_h(x_h, a_h)\}_{(x_h,a_h)\in\mathcal{X}\times\mathcal{A}} \in \mathbb{R}^{XA}_{\geq 0}$.
    Compute matrix loss $M^t = \ell^t(\mu^t)^\top \in \mathbb{R}^{XA\times XA}_{\geq 0}$.
    For each $x_ga_g \in \mathcal{X}\times\mathcal{A}$, from the reverse order of $x_h$, compute $m^{t+1}_{x_ga_g,h}(a_h|x_h)$ and $\widetilde{F}^t_{x_ga_g,x_h}$
        $m^{t+1}_{x_ga_g,h}(a_h|x_h) \propto_{a_h} m^t_{x_ga_g,h}(a_h|x_h)\exp\Big\{-\eta M^t_{x_ha_h,x_ga_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^t_{x_ga_g,x_{h+1}}\Big\},$
        $\widetilde{F}^t_{x_ga_g,x_h} = \log\sum_{a_h\in\mathcal{A}} m^t_{x_ga_g,h}(a_h|x_h)\exp\Big\{-\eta M^t_{x_ha_h,x_ga_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^t_{x_ga_g,x_{h+1}}\Big\},$
    Compute $\lambda^{t+1}_{x_ga_g}$ as
        $\lambda^{t+1}_{x_ga_g} \propto_{x_ga_g} \lambda^t_{x_ga_g}\exp\Big\{-\eta\langle I - E_{\succeq x_ga_g}, M^t\rangle + \widetilde{F}^t_{x_ga_g,x_g}\Big\}.$
```

update the policy in each iteration.

We now prove that Algorithm 24 and Algorithm 13 are actually equivalent.

Lemma 140. *Given the same sequence of M^t, Algorithm 13 and Algorithm 24 outputs the same λ^t and m^t and thus the same ϕ^t.*

Proof. We only need to prove for any x_ga_g, x_h and t, $F^t_{x_ga_g,x_h} = \sum_{s=1}^t \widetilde{F}^s_{x_ga_g,x_h}$. Then λ^t and m^t will be the same in Algorithm 13 and Algorithm 24.

We prove the above claim by induction. For the base case, the claim clearly holds if $h = H+1$ or $t = 1$ by definition. Assume this holds at $t-1$ and $h+1$, then at the h-th step in Algorithm 24,

$$\widetilde{F}^t_{x_ga_g,x_h} = \log\sum_{a_h\in\mathcal{A}} m^t_{x_ga_g,h}(a_h|x_h)\exp\Big\{-\eta M^t_{x_ha_h,x_ga_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^t_{x_ga_g,x_{h+1}}\Big\}$$

$$
\begin{aligned}
&= \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ -\eta \sum_{s=1}^{t} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \Big\} \\
&\quad - \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ -\eta \sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^{t-1}_{x_g a_g, x_{h+1}} \Big\} \\
&= F^t_{x_g a_g, x_h} - F^{t-1}_{x_g a_g, x_h}.
\end{aligned}
$$

Thus $F^t_{x_g a_g, x_h} = \sum_{s=1}^t \widetilde{F}^s_{x_g a_g, x_h}$. We completes the proof by noticing at $H+1$ step, $F^t_{x_g a_g, x_{H+1}} = \widetilde{F}^t_{x_g a_g, x_{H+1}} = 0$. □

F.1.2 Proof of Lemma 50

By Line 6 of Algorithm 9, we have

$$
p^{t+1}_\phi = \frac{p^t_\phi \cdot \exp\{-\eta \langle \phi \mu^t, \ell^t \rangle\}}{\sum_{\phi'} p^t_{\phi'} \cdot \exp\{-\eta \langle \phi' \mu^t, \ell^t \rangle\}} = \frac{p^t_\phi \cdot \exp\{-\eta \langle \phi, M^t \rangle\}}{\sum_{\phi'} p^t_{\phi'} \cdot \exp\{-\eta \langle \phi', M^t \rangle\}}. \tag{F.2}
$$

Repeating this update and using the uniform initialization, we have

$$
p^{t+1}_\phi = \frac{\exp\{-\eta \langle \phi, \sum_{s=1}^t M^s \rangle\}}{\sum_{\phi'} \exp\{-\eta \langle \phi', \sum_{s=1}^t M^s \rangle\}}.
$$

As a result, we have

$$
\phi^t = \sum_\phi p^t_\phi \phi = \frac{\sum_\phi \exp\{-\eta \langle \phi, \sum_{s=1}^{t-1} M^s \rangle\} \phi}{\sum_\phi \exp\{-\eta \langle \phi, \sum_{s=1}^{t-1} M^s \rangle\}} = -\nabla F^{\Phi_0}\left(\eta \sum_{s=1}^{t-1} M^s \right). \tag{F.3}
$$

This proves the lemma. □

F.1.3 Proof of Lemma 51

For any $x_h \succeq x_g$, we define $F_{x_g a_g, x_h}(M)$ by

$$
F_{x_g a_g, x_h}(M) := \log \sum_{m_{x_g a_g} \in \mathcal{V}^{x_h}} \exp(-\langle m_{x_g a_g} e^\top_{x_g a_g}, M \rangle).
$$

Note that for any $\phi \in \Phi_0^{\mathsf{Tr}}$, there exists a unique $(g, x_g, a_g, m_{x_g a_g}) \in [H] \times \mathcal{X} \times \mathcal{A} \times$

$\mathcal{V}^{x_g}$ such that $\phi = \phi_{x_g a_g \to m_{x_g a_g}}$. As a consequence, we have

$$\begin{aligned} F^{\mathsf{EFCE}}(M) &= \log \sum_{\phi \in \Phi_0^{\mathsf{EFCE}}} \exp(-\langle \phi, M \rangle) \\ &= \log \sum_{g, x_g, a_g} \sum_{m_{x_g a_g} \in \mathcal{V}^{x_g}} \exp(-\langle \phi_{x_g a_g \to m_{x_g a_g}}, M \rangle) \\ &= \log \sum_{g, x_g, a_g} \sum_{m_{x_g a_g} \in \mathcal{V}^{x_g}} \exp(-\langle I - E_{\succeq x_g a_g} + m_{x_g a_g} e_{x_g a_g}^\top, M \rangle) \\ &= \log \sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M \rangle + F_{x_g a_g, x_g}(M) \Big\}. \end{aligned}$$

It remains to evaluate $F_{x_g a_g, x_h}(M)$ recurrently, which is handled by the structure of $\mathcal{V}^{x_h}$ as follows:

$$\begin{aligned} & F_{x_g a_g, x_h}(M) \\ &= \log \sum_{m_{x_g a_g} \in \mathcal{V}^{x_h}} \exp(-\langle m_{x_g a_g} e_{x_g a_g}^\top, M \rangle) \\ &= \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \sum_{m_{x_{h+1} a_{h+1}} \in \mathcal{V}^{x_{h+1}}} \exp(-\langle m_{x_g a_g} e_{x_g a_g}^\top, M \rangle) \Big\} \\ &= \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M) \Big\}. \end{aligned}$$

This proves Eq. (7.3) and (7.4).

Calculating the gradient, we have

$$\begin{aligned} & - \nabla F^{\mathsf{EFCE}}(M) \\ &= \frac{\sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M \rangle + F_{x_g a_g, x_g}(M) \Big\} \big[I - E_{\succeq x_g a_g} - \nabla F_{x_g a_g, x_h}(M) \big]}{\sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M \rangle + F_{x_g a_g, x_g}(M) \Big\}} \\ &= \sum_{g, x_g, a_g} \lambda_{x_g, a_g} \big[I - E_{\succeq x_g a_g} - \nabla F_{x_g a_g, x_h}(M) \big]. \end{aligned} \tag{F.4}$$

It remains to compute $\nabla F_{x_g a_g, x_h}(M)$. By the recurrent formula, we have

$$- \nabla F_{x_g a_g, x_h}(M)$$

$$= \frac{\sum_{a_h \in \mathcal{A}} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M) \Big\} \big[e_{x_h a_h} e_{x_g a_g}^\top - \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \nabla F_{x_g a_g, x_{h+1}}(M) \big]}{\sum_{a_h \in \mathcal{A}} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M) \Big\}}$$

$$= \sum_{a_h \in \mathcal{A}} m_{x_g a_g, h}(a_h | x_h) \big[e_{x_h a_h} e_{x_g a_g}^\top + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} (-\nabla F_{x_g a_g, x_{h+1}})(M) \big].$$

This gives a recursion formula for $-\nabla F_{x_g a_g, x_h}(M)$. Solving this recursion formula, we get

$$-\nabla F_{x_g a_g, x_h}(M) = m_{x_g a_g} e_{x_g a_g}^\top,$$

Plugging this into Eq. (F.4) completes the proof. □

F.1.4 Runtime of Algorithm 13

Here we explain how Lemma 51 and its execution in Algorithm 13 is an $O(X^2 A^2)$ time (in floating-point operations) efficient implementation of $-\nabla F^{\mathsf{Tr}}(M)$ for any matrix $M \in \mathbb{R}^{XA \times XA}$.

First, the function value $F^{\mathsf{Tr}}(M)$ can be recursively evaluated using (7.3) & (7.4), where we first evaluate (7.4) for any $x_g a_g \in \mathcal{X} \times \mathcal{A}$ recursively in a bottom-up fashion over $\{x_h : x_h \succeq x_g\}$ (i.e. the subtree rooted at x_g) up until $x_h = x_g$, and then plug in the resulting values of $F_{x_g a_g, x_g}(M)$ into (7.3) to obtain $F^{\mathsf{Tr}}(M)$. This process costs $O(XA)$ operations for each $x_g a_g$, so in total costs $O(X^2 A^2)$ operations. Second, (7.5)-(7.7) show that the gradient can be obtained without much extra cost: By (7.7), $-\nabla F^{\mathsf{Tr}}(M)$ is determined by the parameters (λ, m), which then by (7.5) & (7.6) are exactly the ratios of the recursive log-sum-exps which we already evaluated in the previous step, and thus can be directly yielded (with cost of the same-order) while evaluating $F^{\mathsf{Tr}}(M)$. So the total runtime of the recursive computations in Lemma 51 (i.e. Algorithm 13) is $O(X^2 A^2)$.

F.1.5 Proof of Lemma 53

We check solving optimization problem (7.13) will result in exactly the same form of Algorithm 13. The OMD form (7.14) is similar.

Using the definition of $H^{\mathsf{EFCE}}(\lambda, m)$, the objective function in (7.13) can be written as

$$H(\lambda) + \sum_{g, x_g a_g} \lambda_{x_g a_g} \left[\eta \langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s \rangle + \eta \langle m_{x_g a_g}, \sum_{s=1}^{t} M^s_{\cdot, x_g a_g} \rangle + H_{x_g}(m_{x_g a_g}) \right].$$

First fix λ. and consider $m_{x_g a_g}$, which is just to minimize $\eta \langle m_{x_g a_g}, \sum_{s=1}^{t} M^s_{\cdot, x_g a_g} \rangle + H_{x_g}(m_{x_g a_g})$.

This is similar to form studied in Appendix B of Kozuno et al. [2021] (or see Lemma 157 for a full proof), which implies that, the optimum is achieved at

$$m^{t+1}_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp \Big\{ \sum_{s=1}^{t} \big[-\eta M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \big] \Big\},$$

where

$$F^t_{x_g a_g, x_h} = \log \sum_{a_h \in \mathcal{A}} \exp \Big\{ \sum_{s=1}^{t} \big[-\eta M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \big] \Big\}.$$

Plug in the optimal $m_{x_g a_g}$, the object now becomes

$$H(\lambda) + \sum_{g, x_g a_g} \lambda_{x_g a_g} \left[\eta \langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s \rangle - F^t_{x_g a_g, x_g} \right].$$

This is a standard KL-regularized linear optimization problem on simplex. The optimum is achieved at

$$\lambda^{t+1}_{x_g a_g} \propto_{x_g a_g} \exp \Big\{ -\eta \langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s \rangle + F^t_{x_g a_g, x_g} \Big\}.$$

This gives the update of λ^{t+1} and m^{t+1} as in Algorithm 13. This completes the proof. □

F.2 Proofs for Section 7.1.3

F.2.1 Proof of Theorem 54

Using regret bound of Φ-Hedge algorithm (Lemma 118), we get

$$\begin{aligned}\mathrm{Reg}^{\mathsf{Tr}}(T) &\le \frac{\log|\Phi_0^{\mathsf{EFCE}}|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}} p_\phi^t\big(\langle \phi\mu^t, \ell^t\rangle\big)^2 \\ &\overset{(i)}{\le} \frac{\log|\Phi_0^{\mathsf{EFCE}}|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}} p_\phi^t H^2 = \frac{\log|\Phi_0^{\mathsf{EFCE}}|}{\eta} + \frac{\eta H^2 T}{2}.\end{aligned}$$

Here, (i) uses $\langle\phi\mu^t, \ell^t\rangle \in [0, H]$. Note that $\Phi_0^{\mathsf{EFCE}} := \bigcup_{g, x_g a_g}\bigcup_{v^{x_g}\in\mathcal{V}^{x_g}}\big\{\phi_{x_g a_g \to v^{x_g}}\big\}$ has cardinality upper bounded by $|\Phi_0^{\mathsf{EFCE}}| \le XA^{\|\Pi\|_1+1}$ by Lemma 114. Substitute this into the regret bound, we have

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{\log(XA^{\|\Pi\|_1+1})}{\eta} + \frac{\eta H^2 T}{2} \le \frac{2\|\Pi\|_1\iota}{\eta} + \frac{\eta H^2 T}{2},$$

where $\iota = \log(XA)$ is a log-term. Choosing $\eta = 2\sqrt{\|\Pi\|_1\iota/(H^2T)}$ gives $\mathrm{Reg}^{\mathsf{Tr}}(T) \le 2\sqrt{H^2\|\Pi\|_1\iota T}$, which completes the proof. □

F.2.2 Proof of Theorem 55

For the loss estimator $\widetilde{\ell}$, we have the following lemma (see Lemma 3 in Kozuno et al. [2021]):

Lemma 141. *Let $\delta \in (0,1)$ and $\gamma \in (0,\infty)$. Fix $h \in [H]$, and let $\alpha(x_h, a_h) \in [0, 2\gamma]$ be some constant for each $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$. Then with probability at least $1-\delta$, we have*

$$\sum_{t=1}^{T}\sum_{x_h\in\mathcal{X}_h, a_h\in\mathcal{A}} \alpha(x_h, a_h)\Big(\widetilde{\ell}_h^t(x_h, a_h) - \ell_h^t(x_h, a_h)\Big) \le \log\frac{1}{\delta}.$$

Proof of Theorem 55. We have

$$\begin{aligned}
\mathrm{Reg}^{\mathsf{Tr}}(T) &= \max_{\phi\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \mu^t - \phi\mu^t, \ell^t\rangle \\
&\le \underbrace{\sum_{t=1}^{T}\langle \mu^t, \ell^t - \widetilde{\ell}^t\rangle}_{\mathrm{BIAS}_1} + \underbrace{\max_{\phi\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \phi\mu^t, \widetilde{\ell}^t - \ell^t\rangle}_{\mathrm{BIAS}_2} + \underbrace{\max_{\phi\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \mu^t - \phi\mu^t, \widetilde{\ell}^t\rangle}_{\mathrm{REGRET}}.
\end{aligned}$$

We use the following three lemmas to bound the terms above respectively. In these lemmas, $\iota = \log(3XA/\delta)$ is a log factor.

Lemma 142 (Bound on BIAS_1)**.** *With probability at least* $1-\delta/3$*, we have*

$$\mathrm{BIAS}_1 \le H\sqrt{2T\iota} + \gamma XAT.$$

Lemma 143 (Bound on BIAS_2)**.** *With probability at least* $1-\delta/3$*, we have*

$$\mathrm{BIAS}_2 \le \|\Pi\|_1\iota/\gamma.$$

Lemma 144 (Bound on REGRET)**.** *With probability at least* $1-\delta/3$*, we have*

$$\mathrm{REGRET} \le \log|\Phi_0^{\mathsf{EFCE}}|/\eta + \eta HXAT + \eta HXA\iota/\gamma..$$

Lemma 142, 143, and 144 bound bias terms and regret term respectively. Using these lemmas, we have with probability at least $1-\delta$,

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{\log|\Phi_0^{\mathsf{EFCE}}|}{\eta} + \eta HXAT + \eta HXA\iota/\gamma + \|\Pi\|_1\iota/\gamma + H\sqrt{2T\iota} + \gamma XAT.$$

Because $|\Phi_0^{\mathsf{EFCE}}| \le XA^{\|\Pi\|_1+1}$, we further have

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{2\|\Pi\|_1\iota}{\eta} + \eta HXAT + \eta HXA\iota/\gamma + \|\Pi\|_1\iota/\gamma + H\sqrt{2T\iota} + \gamma XAT.$$

Choosing $\gamma = \sqrt{\|\Pi\|_1\iota/(XAT)}$ and $\eta = \sqrt{\|\Pi\|_1\iota/(HXAT)}$ gives

$$\begin{aligned}\mathrm{Reg}^{\mathsf{Tr}}(T) &\le 5\sqrt{HXA\|\Pi\|_1T\iota} + XA\iota\sqrt{H} + H\sqrt{2T\iota}\\ &\le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota\cdot T} + XA\iota\sqrt{H}),\end{aligned}$$

where we uses $\|\Pi\|_1 \ge H$. Notice that there is a "trivial" bound $\mathrm{Reg}^{\mathsf{Tr}}(T) \le HT$. For $T \ge XA\iota/\|\Pi\|_1$, we have $XA\iota\sqrt{H} \le \sqrt{HXA\|\Pi\|_1\iota\cdot T}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota\cdot T})$; For $T \le XA\iota/\|\Pi\|_1$, we have $HT \le \sqrt{HXA\|\Pi\|_1\iota\cdot T}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le HT \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota\cdot T})$. Therefore, we always have

$$\mathrm{Reg}^{\mathsf{Tr}} \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota\cdot T}).$$

This completes the proof. □

Here, we give the proofs of the lemmas we used above.

Proof of Lemma 142. We further decompose BIAS_1 to two terms by

$$\mathrm{BIAS}_1 = \sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \widetilde{\ell}^t\right\rangle = \underbrace{\sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\}\right\rangle}_{(A)} + \underbrace{\sum_{t=1}^{T}\left\langle \mu^t, \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\} - \widetilde{\ell}^t\right\rangle}_{(B)}.$$

To bound (A), plug in the definition of loss estimator,

$$\begin{aligned}&\sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\}\right\rangle\\ =& \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h}\mu^t_{1:h}(x_h,a_h)\left[\ell^t_h(x_h,a_h) - \frac{\mu^t_{1:h}(x_h,a_h)\ell^t_h(x_h,a_h)}{\mu^t_{1:h}(x_h,a_h)+\gamma}\right]\\ =& \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h}\mu^t_{1:h}(x_h,a_h)\ell^t_h(x_h,a_h)\left[\frac{\gamma}{\mu^t_{1:h}(x_h,a_h)+\gamma}\right]\\ \le& \gamma\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h}\ell^t_h(x_h,a_h) \le \gamma XAT,\end{aligned}$$

where the last inequality is by $\ell^t_h(x_h,a_h) \in [0,1]$.

To bound (B), first notice

$$\begin{aligned}\left\langle \mu^t, \widetilde{\ell}^t \right\rangle &= \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}^t(x_h, a_h) \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma} \\ &\leq \sum_{h=1}^{H} \sum_{x_h, a_h} \mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\} = \sum_{h=1}^{H} 1 = H.\end{aligned}$$

Then by Azuma-Hoeffding, with probability at least $1 - \delta/3$, we have

$$\sum_{t=1}^{T} \left\langle \mu^t, \mathbb{E}\left\{\widetilde{\ell}^t | \mathcal{F}^{t-1}\right\} - \widetilde{\ell}^t \right\rangle \leq H\sqrt{2T \log(3/\delta)} \leq H\sqrt{2T\iota}.$$

Combining the bounds for (A) and (B) gives the desired result. □

Proof of Lemma 143. We have with probability at least $1 - \delta/3$,

$$\begin{aligned}\mathrm{BIAS}_2 &= \max_{\phi \in \Phi^{\mathsf{EFCE}}} \sum_{t=1}^{T} \left\langle \phi\mu^t, \widetilde{\ell}^t - \ell^t \right\rangle \\ &= \max_{\phi \in \Phi^{\mathsf{EFCE}}} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h, a_h} (\phi\mu^t)_{1:h}(x_h, a_h) \left[\widetilde{\ell}_h^t(x_h, a_h) - \ell_h^t(x_h, a_h)\right] \\ &= \max_{\phi \in \Phi^{\mathsf{EFCE}}} \sum_{h=1}^{H} \sum_{x_h, a_h} \frac{(\phi\mu^t)_{1:h}(x_h, a_h)}{\gamma} \sum_{t=1}^{T} \gamma \left[\widetilde{\ell}_h^t(x_h, a_h) - \ell_h^t(x_h, a_h)\right] \\ &\overset{(i)}{\leq} \frac{\log(3XA/\delta)}{\gamma} \max_{\phi \in \Phi^{\mathsf{EFCE}}} \sum_{h=1}^{H} \sum_{x_h, a_h} (\phi\mu^t)_{1:h}(x_h, a_h) \\ &\leq \|\Pi\|_1 \iota/\gamma,\end{aligned}$$

where (i) is by applying Lemma 141 for each (x_h, a_h) pair. To be more specific, we choose $\alpha(x_h', a_h') = \gamma \mathbf{1}\left\{(x_h', a_h') = (x_h, a_h)\right\}$, then Lemma 141 yields that

$$\sum_{t=1}^{T} \gamma \left[\widetilde{\ell}_h^t(x_h, a_h) - \ell_h^t(x_h, a_h)\right] \leq \log \frac{3XA}{\delta}$$

with probability at least $1 - \delta/(3XA)$. Then taking union bound gives the inequality in (i). □

Proof of Lemma 144. Note that by Lemma 118, we have

$$\mathrm{REGRET} \le \frac{\log |\Phi_0^{\mathsf{Tr}}|}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{\phi \in \Phi_0^{\mathsf{Tr}}} p_\phi^t (\langle \phi\mu^t, \widetilde{\ell}^t \rangle)^2.$$

To bound the second term, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{\phi \in \Phi_0^{\mathsf{Tr}}} p_\phi^t (\langle \phi\mu^t, \widetilde{\ell}^t \rangle)^2 \\
&\le 2 \sum_{h' \ge h} \sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h)} \sum_{\phi \in \Phi_0^{\mathsf{Tr}}} \frac{p_\phi^t (\phi\mu^t)_{1:h}(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h) + \gamma} \widetilde{\ell}_{h'}^t(x_{h'}, a_{h'}) \\
&\le 2 \sum_{h' \ge h} \sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h)} \sum_{\phi \in \Phi_0^{\mathsf{Tr}}} \frac{p_\phi^t (\phi\mu^t)_{1:h}(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h)} \widetilde{\ell}_{h'}^t(x_{h'}, a_{h'}) \\
&\overset{(i)}{=} 2 \sum_{h' \ge h} \sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h)} \widetilde{\ell}_{h'}^t(x_{h'}, a_{h'}) \\
&\overset{(ii)}{\le} 2 \sum_{h' \ge h} \Big(\sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h)} \ell_{h'}^t(x_{h'}, a_{h'}) + X_h A \iota / \gamma \Big) \\
&\le 2HXAT + 2HXA\iota/\gamma,
\end{aligned}$$

where (i) uses that μ^t is the solution of the fixed point equation $\mu^t = \sum_{\phi \in \Phi_0^{\mathsf{Tr}}} p_\phi^t \phi\mu^t$; (ii) is by Lemma 141, which gives

$$\sum_{t=1}^{T} \sum_{(x_{h'}, a_{h'}) \in \mathcal{C}_{h'}(x_h, a_h)} \gamma \Big(\widetilde{\ell}_{h'}^t(x_{h'}, a_{h'}) - \ell_{h'}^t(x_{h'}, a_{h'}) \Big) \le \log \frac{3XA}{\delta}$$

with probability at least $1 - \delta/(3XA)$ (choosing $\alpha(x_h', a_h') = \gamma \mathbf{1}\{(x_h', a_h') \in \mathcal{C}_{h'}(x_h, a_h)\}$ in the lemma). Then taking union bound yields that (ii) holds with probability at least $1 - \delta/3$.

Finally, putting everything together, the lemma is proved. □

F.3 Proof of Theorem 58

Here we restate the theorem for convenience.

Theorem 145 (Sample complexity under bandit feedback)**.** *Run Balanced EFCE-OMD (Algorithm 14) with $\eta = \sqrt{XA\iota/H^4T}$ and $\gamma = 2\sqrt{XA\iota/H^2T}$. Then with probability at least $1-\delta$, we have the following extensive-form trigger regret bound,*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le 200\sqrt{XAH^4T\iota},$$

where $\iota = \log(10HXA/\delta)$ is the log factor.

Proof. By the fixed point property of our algorithm, we have the regret decomposition

$$\begin{aligned}
&\mathrm{Reg}^{\mathsf{Tr}}(T)\\
&= \sup_{\phi^\star\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \mu^t - \phi^\star\mu^t, \ell^t\rangle\\
&= \sup_{\phi^\star\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \phi^t\mu^t - \phi^\star\mu^t, \ell^t\rangle\\
&= \sup_{\phi^\star\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \phi^t - \phi^\star, \ell^t(\mu^t)^\top\rangle\\
&\le \underbrace{\sup_{\phi^\star\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \phi^t - \phi^\star, \widetilde{M}^t\rangle}_{\widetilde{\mathrm{REGRET}}^{\mathsf{EFCE}}(T)} + \underbrace{\sum_{t=1}^{T}\langle \phi^t, \ell^t(\mu^t)^\top - \widetilde{M}^t\rangle}_{\mathrm{BIAS}_1} + \underbrace{\sup_{\phi^\star\in\Phi^{\mathsf{EFCE}}}\sum_{t=1}^{T}\langle \phi^\star, \widetilde{M}^t - \ell^t(\mu^t)^\top\rangle}_{\mathrm{BIAS}_2}.
\end{aligned}$$

We bound the term $\widetilde{\mathrm{REGRET}}^{\mathsf{EFCE}}(T)$, BIAS_1, and BIAS_2 in the following lemmas, whose proofs are presented in Section F.3.2 and F.3.3.

Lemma 146 (Bound on $\widetilde{\mathrm{REGRET}}^{\mathsf{EFCE}}(T)$)**.** *Assume that $\gamma \ge 2\eta H$. We have with probability at least $1-\delta/3$ that*

$$\widetilde{\mathrm{REGRET}}^{\mathsf{EFCE}}(T) \le \frac{XA\log(XA^2)}{\eta} + 22\eta H^4T + \frac{38\eta H^3XA\iota}{\gamma},$$

where $\iota = \log(10HXA/\delta)$ is the log factor.

Lemma 147 (Bound on BIAS_1)**.** *We have with probability at least* $1 - \delta/3$ *that*

$$\mathrm{BIAS}_1 \le 2\gamma H^2 T + 2H\sqrt{T}\iota,$$

where $\iota = \log(3/\delta)$ *is the log factor.*

Lemma 148 (Bound on BIAS_2)**.** *We have with probability at least* $1 - \delta/3$ *that*

$$\mathrm{BIAS}_2 \le \frac{XA\iota}{\gamma},$$

where $\iota = \log(3XA/\delta)$ *is the log factor.*

By these three lemmas, whenever $\gamma \ge 2\eta H$, we have

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{XA\log(XA^2)}{\eta} + 22\eta H^4 T + \frac{38\eta H^3 XA\iota}{\gamma} + 2\gamma H^2 T + 2H\sqrt{T}\iota + \frac{XA\iota}{\gamma}.$$

Taking $\eta = \sqrt{XA\iota/H^4T}$ and $\gamma = 2\sqrt{XA\iota/H^2T}$, we get

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le 100\Big[\sqrt{XAH^4T\iota} + H^2XA\iota\Big].$$

Notice that there is the "trivial" bound $\mathrm{Reg}^{\mathsf{Tr}}(T) \le HT$. For $T \ge XA\iota$, we have $H^2XA\iota \le \sqrt{XAH^4T\iota}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le 200\sqrt{H^4XAT\iota}$; For $T \le XA\iota$, we have $HT \le \sqrt{XAH^4T\iota}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le HT \le \sqrt{H^4XAT\iota}$. Therefore, we always have

$$\mathrm{Reg}^{\mathsf{Tr}} \le 200\sqrt{H^4XAT\iota}.$$

This gives the desired bound. □

The rest of this section is organized as follows. We introduce some notations in Section F.3.1. In Section F.3.2, we bound the regret term $\widetilde{\mathrm{REGRET}}^{\mathsf{EFCE}}(T)$. In Section F.3.3, we bound the two bias terms BIAS_1 and BIAS_2.

F.3.1 Some preparations

Note that $m^t_{x_g a_g} \in \Pi^{x_g}$ is a subtree policy rooted at x_g, we denote

$$m^t_{x_g a_g, g:h}(x_h, a_h) := \prod_{h'=g}^{h} m^t_{x_g a_g, h'}(a'_h | x'_h),$$

where $(x_g, a_g, a_{g+1}, a_{g+1}, \cdots, x_{h-1}, a_{h-1})$ is the unique history leading to (x_h, a_h).

Note that we have $\phi^t = \sum_{g, x_g, a_g} \lambda^t_{x_g a_g}(I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g})$ and $\phi^t \mu^t = \mu^t$. These two equations give

$$\sum_{g, x_g, a_g} \lambda^t_{x_g a_g} E_{\succeq x_g a_g} \mu^t = \sum_{g, x_g, a_g} \lambda^t_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g} \in \mathbb{R}^{XA}. \tag{F.5}$$

As a consequence, for any $x_g a_g$, we have

$$\lambda^t_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g} \le \sum_{g, x_g, a_g} \lambda^t_{x_g a_g} \mu^t = \mu^t. \tag{F.6}$$

Here $\lambda^t_{x_g a_g} \in \Delta_{XA}, \mu^t_{x_g a_g} = \mu^t_{1:g}(x_g, a_g)$ are two scalers, and $m^t_{x_g a_g} \in \Pi^{x_g}$ and $\mu^t \in \Pi$ are two vectors of length XA. The $\le$ above is understood in an entrywise sense.

We also define (recall that $\{p^t_h\}_{h \in \{0\} \cup [H], t \ge 1}$ are the adversarial probability transition function)

$$p^t(x_h) := p^t_0(x_1) \prod_{h'=1}^{h-1} p^t_{h'}(x_{h'+1} | x_{h'}, a_{h'}). \tag{F.7}$$

Note that $p^t(x_h) \in [0, 1]$. Furthermore, for any policy $\mu \in \Pi$ and any (h, t), we have

$$\sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) p^t(x_h, a_h) = 1, \tag{F.8}$$

as the left-hand side is the probability of visiting some (x_h, a_h) in episode t using policy μ.

F.3.2 Proof of Lemma 146

Recall that $\widetilde{\text{REGRET}}^{\textsf{EFCE}}(T)$ is defined as

$$\widetilde{\text{REGRET}}^{\textsf{EFCE}}(T) := \sup_{\phi^\star \in \Phi^{\textsf{EFCE}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \widetilde{M}^t \rangle.$$

First, we claim that

$$\sup_{\phi^\star \in \Phi^{\textsf{EFCE}}} \langle -\phi^\star, M \rangle = \sup_{\phi^\star \in \Phi_0^{\textsf{EFCE}}} \langle -\phi^\star, M \rangle \le \frac{1}{\eta} F_{\textsf{bal}}^{\textsf{EFCE}}(M).$$

for any $M \in \mathbb{R}^{XA \times XA}$. Indeed, the first equation follows from $\Phi^{\textsf{EFCE}} = \text{conv}\big\{\Phi_0^{\textsf{Tr}}\big\}$. The inequality is due to the following argument: for any fixed M, the maximizer $\phi_{x_g a_g \to m_{x_g a_g}} \in \Phi_0^{\textsf{EFCE}}$ specifies a trigger sequence $x_g a_g$ and a deterministic subtree policy $m_{x_g a_g}$ starting from x_g. Replacing all the sums by this realization in the formula of $F_{\textsf{bal}}^{\textsf{EFCE}}$ (c.f. Eq. (7.16)) and $F^\star_{x_g a_g, x_h}$ (c.f. Eq. (7.17)) exactly gives $\langle -\phi_{x_g a_g \to m_{x_g a_g}}, M \rangle = \sup_{\phi^\star \in \Phi_0^{\textsf{EFCE}}} \langle -\phi^\star, M \rangle$. This proves the claim.

This claim gives

$$\begin{aligned}
&\widetilde{\text{REGRET}}^{\textsf{EFCE}}(T) \\
&= \sup_{\phi^\star \in \Phi^{\textsf{EFCE}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \widetilde{M}^t \rangle = \sup_{\phi^\star \in \Phi^{\textsf{EFCE}}} \langle -\phi^\star, \sum_{t=1}^{T} \widetilde{M}^t \rangle + \sum_{t=1}^{T} \langle \phi^t, \widetilde{M}^t \rangle \\
&\le \frac{1}{\eta} F_{\textsf{bal}}^{\textsf{EFCE}}\Big(\sum_{t=1}^{T} \widetilde{M}^t\Big) + \sum_{t=1}^{T} \langle \phi^t, \widetilde{M}^t \rangle = \frac{1}{\eta} F_{\textsf{bal}}^{\textsf{EFCE}}(0) + \sum_{t=1}^{T} D^t,
\end{aligned} \tag{F.9}$$

where D^t is given by

$$D^t = \frac{1}{\eta} F_{\textsf{bal}}^{\textsf{EFCE}}\Big(\eta \sum_{s=1}^{t} \widetilde{M}^s\Big) - \frac{1}{\eta} F_{\textsf{bal}}^{\textsf{EFCE}}\Big(\eta \sum_{s=1}^{t-1} \widetilde{M}^s\Big) + \langle \phi^t, \widetilde{M}^t \rangle. \tag{F.10}$$

The following lemma gives bound on the initial entropy $F_{\textsf{bal}}^{\textsf{EFCE}}(0)$ with proof in Section F.3.2.

Lemma 149 (Bound on initial entropy)**.** *We have*

$$F_{\mathsf{bal}}^{\mathsf{EFCE}}(0) = XA \log \sum_{g, x_g, a_g} \exp\{[X_{\succeq x_g} A \log A]/XA\} \le XA \log(XA^2). \tag{F.11}$$

The following lemma gives a reformulation of the stability term D^t with proof in Section F.3.2.

Lemma 150 (Reformulation of stability term via incremental update)**.** *We have*

$$D^t = \overline{F}^t/\eta + \langle \phi^t, \widetilde{M}^t \rangle, \tag{F.12}$$

where we have

$$\overline{F}^t = XA \log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ \frac{1}{XA} \big[-\eta \langle I - E_{\succeq x_g a_g}, \widetilde{M}^t \rangle + F^{\star,t}_{x_g a_g, x_g} \big] \Big\}, \tag{F.13}$$

$$\begin{aligned} F^{\star,t}_{x_g a_g, x_h} = {} & \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h \in \mathcal{A}} m^t_{x_g a_g, h}(a_h | x_h) \exp\Big\{ \mu^{\star,h}_{g:h}(x_h, a_h) \\ & \cdot \big[-\eta \underbrace{\mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g}_h (x_h, a_h)}_{= \widetilde{M}^t_{x_h a_h, x_g a_g}} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^{\star,t}_{x_g a_g, x_{h+1}} \big] \Big\}, \quad \forall (x_h, a_h) \succeq x_g, \end{aligned} \tag{F.14}$$

and (note that $F^{\star}_{x_g a_g, x_g}(\mathbf{0})$ is as defined in Eq. (7.17) by plugging in $M = \mathbf{0}$)

$$\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ \frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0}) + \frac{1}{XA} \sum_{s=1}^{t-1} \Big(-\eta \langle I - E_{\succeq x_g a_g}, \widetilde{M}^s \rangle + F^{\star,s}_{x_g a_g, x_g} \Big) \Big\}, \tag{F.15}$$

$$m^t_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{ \mu^{\star,h}_{g:h}(x_h, a_h) \sum_{s=1}^{t-1} \Big(-\eta \underbrace{\mu^s_{x_g a_g} \widetilde{\ell}^{s, x_g a_g}_h (x_h, a_h)}_{= \widetilde{M}^s_{x_h a_h, x_g a_g}} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^{\star,s}_{x_g a_g, x_{h+1}} \Big) \Big\}. \tag{F.16}$$

To upper bound D^t, note that we have

$$
\begin{aligned}
&D^t = \overline{F}^t/\eta + \langle \phi^t, \widetilde{M}^t \rangle \\
&= \langle \phi^t, \widetilde{M}^t \rangle + \frac{XA}{\eta} \log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ \frac{1}{XA} \big[-\eta \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle + \Delta^t_{x_g a_g} \big] \Big\},
\end{aligned}
$$

where $\Delta^t_{x_g a_g}$ is given by

$$
\begin{aligned}
\Delta^t_{x_g a_g} &:= F^{\star,t}_{x_g a_g, x_h} + \eta \langle m^t_{x_g a_g} \mu^t_{x_g a_g}, \widetilde{\ell}^{t, x_g a_g} \rangle \\
&= F^{\star,t}_{x_g a_g, x_h} + \eta \langle m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle.
\end{aligned}
\tag{F.17}
$$

Note that $-\eta \langle m^t_{x_g a_g} \mu^t_{x_g a_g}, \widetilde{\ell}^{t, x_g a_g} \rangle$ will be the linear term in the Taylor expansion of $F^{\star,t}_{x_g a_g, x_h}$ over variable $\widetilde{\ell}^t$ at 0, so $\Delta^t_{x_g a_g}$ can be understood as the nonlinear part within $F^{\star,t}_{x_g a_g, x_g}$. By convexity of $F^{\star,t}_{x_g a_g, x_h}$ as a function of $\widetilde{\ell}^t$, we have $\Delta^t_{x_g a_g} \geq 0$. Furthermore, we have the following almost sure upper bound of $\sup_{g, x_g a_g} \Delta^t_{x_g a_g}$ with proof in Section F.3.2.

Lemma 151 (Bound on $\sup_{g, x_g a_g} \Delta^t_{x_g a_g}$)**.** *We have for all $t \in [T]$ that, almost surely,*

$$
\frac{1}{XA} \sup_{g, x_g a_g} \Delta^t_{x_g a_g} \leq \frac{2\eta^2}{\gamma^2} H^2.
$$

Given this lemma, we further assume that $\gamma \geq 2H\eta$ so that $\frac{1}{XA} \sup_{g, x_g a_g} \Delta^t_{x_g a_g} \leq 1$. Now we use elementary inequalities $\log(1+x) \leq x$ and

$$
e^{-x+c} \leq 1 - (x-c) + \frac{e}{2}(x-c)^2 \leq 1 - (x-c) + e(x^2 + c^2), \qquad \forall x \geq 0, c \leq 1,
$$

and (taking $c = \Delta^t_{x_g a_g}$ for each $(g, x_g a_g)$ below) we get

$$
\begin{aligned}
&\log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ \frac{1}{XA} \big[-\eta \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle + \Delta^t_{x_g a_g} \big] \Big\} \\
&\leq \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ \frac{1}{XA} \big[-\eta \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle + \Delta^t_{x_g a_g} \big] \Big\} - 1 \\
&\leq \Big\{ \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \Big(1 + \frac{1}{XA} \big[-\eta \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle + \Delta^t_{x_g a_g} \big]
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \frac{e}{X^2A^2}\big[\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\big]\Big\} - 1 \\
&= -\frac{\eta}{XA}\langle \phi^t, \widetilde{M}^t\rangle + \frac{1}{XA}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Delta^t_{x_g a_g} \\
&\quad + \frac{e}{X^2A^2}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Big(\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\Big).
\end{aligned}
$$

This gives that

$$
\begin{aligned}
&D^t \\
&\le \frac{1}{\eta}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Delta^t_{x_g a_g} + \frac{e}{\eta XA}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Big(\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\Big) \\
&\overset{(i)}{\le} \frac{1}{\eta}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Delta^t_{x_g a_g}\left(1 + \frac{e}{XA}\sup_{g, x_g a_g}\Delta^t_{x_g a_g}\right) + \frac{e\eta}{XA}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 \\
&\overset{(ii)}{\le} \underbrace{\frac{4}{\eta}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Delta^t_{x_g a_g}}_{\mathrm{I}_t} + \underbrace{\frac{e\eta}{XA}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2}_{\mathrm{II}_t}.
\end{aligned}
\tag{F.18}
$$

In the line of inequality above, (i) used $\Delta^t_{x_g a_g} \ge 0$, and (ii) used Lemma 151 and $\gamma \ge 2\eta H$.

Next, we use the following lemmas to bound $\sum_{t=1}^T \mathrm{I}_t$ and $\sum_{t=1}^T \mathrm{II}_t$, with proofs in Section F.3.2 and F.3.2.

Lemma 152 (Bound on $\sum_{t=1}^T \mathrm{I}_t$)**.** *With probability at least $1 - \delta/10$, we have*

$$
\sum_{t=1}^T \mathrm{I}_t \le 16\eta H^3 T + \frac{32\eta H^3 XA\iota}{\gamma},
$$

where $\iota := \log(10H/\delta)$ is the log factor.

Lemma 153 (Bound on $\sum_{t=1}^T \mathrm{II}_t$)**.** *With probability at least $1 - \delta/10$, we have*

$$
\sum_{t=1}^T \mathrm{II}_t \le 6\eta HT + \frac{6\eta HXA\iota}{\gamma},
$$

where $\iota := \log(10XA/\delta)$ is the log factor.

Combining Eq. (F.9), Lemma 149 and 150, Eq. (F.18), and Lemma 152 and 153, we have

$$\widetilde{\text{REGRET}}^{\mathsf{EFCE}}(T) \le \frac{XA\log(XA^2)}{\eta} + 22\eta H^4 T + \frac{38\eta H^3 XA\iota}{\gamma}$$

with probability at least $1-\delta/3$, where $\iota := \log(10XAH/\delta)$ is the log factor. This completes the proof of Lemma 146.

Proof of Lemma 149

Proof of Lemma 149. By the definition of balanced EFCE log-partition function (see (7.16) and (7.17)), we have

$$F^{\mathsf{Tr}}_{\mathsf{bal}}(\mathbf{0}) = XA\log\sum_{g,x_g,a_g}\exp\Big\{\frac{1}{XA}\big[F^{\star}_{x_g a_g, x_g}(\mathbf{0})\big]\Big\},$$

where for any $x_h \succeq x_g$,

$$F^{\star}_{x_g a_g, x_h}(\mathbf{0}) = \frac{1}{\mu^{\star,h}_{g:h}(x_h,a_h)}\log\sum_{a_h}\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)}F^{\star}_{x_g a_g, x_{h+1}}(\mathbf{0})\big]\Big\}. \tag{F.19}$$

So we only need to prove that $F^{\star}_{x_g a_g, x_g}(\mathbf{0}) = X_{\succeq x_g}A\log A$. In fact, we can use backward induction to prove the following: for any $x_g\in\mathcal{X}_g$ and $x_h\in\mathcal{C}_h(x_g)$, we have

$$F^{\star}_{x_g a_g, x_h}(\mathbf{0}) = \sum_{h'=h}^{H}\frac{\mathcal{C}_{h'}(x_h)}{\mu^{\star:h'}_{g:h-1}(x_{h-1},a_{h-1})}A\log A, \tag{F.20}$$

(with convention $\mu^{\star,h'}_{g:g-1}=1$) where x_{h-1},a_{h-1} is uniquely determined as $x_g \preceq (x_{h-1},a_{h-1}) \prec x_h$. It is easy to see that, choosing $x_h = x_g$ in (F.20) gives $F^{\star}_{x_g a_g, x_g}(\mathbf{0}) = X_{\succeq x_g}A\log A$.

Next we prove (F.20). We use backward induction on h. When $h = H$, from

(F.19), for any $x_H \in \mathcal{C}_H(x_g)$, we have

$$F^{\star}_{x_g a_g, x_H}(\mathbf{0}) = \frac{1}{\mu^{\star,H}_{H}(a_H|x_H)} \log A = A \log A.$$

Now suppose (F.20) is true for $h+1$ for an $h \in [g, H-1]$. By the recursive formula and the induction hypothesis, for any x_h, we have

$$\begin{aligned}
&F^{\star}_{x_g a_g, x_h}(\mathbf{0}) \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^{\star}_{x_g a_g, x_{h+1}}(\mathbf{0})\big]\Big\} \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h}(x_h, a_h)} A \log A\big]\Big\}.
\end{aligned}$$

Then by the definition of the balanced policies (6.1), we have

$$\begin{aligned}
&\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \frac{|\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h}(x_h, a_h)} \\
&= \frac{\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} |\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} \cdot \frac{|\mathcal{C}_{h'}(x_h)|}{|\mathcal{C}_{h'}(x_h, a_h)|} = \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})},
\end{aligned}$$

which is also independent of a_h. So we have

$$\begin{aligned}
&F^{\star}_{x_g a_g, x_h}(\mathbf{0}) \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \Bigg\{A \cdot \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A\big]\Big\}\Bigg\} \\
&= \frac{\log A}{\mu^{\star,h}_{g:h}(x_h, a_h)} + \sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A \\
&= \sum_{h'=h}^{H} \frac{\mathcal{C}_{h'}(x_h)}{\mu^{\star:h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A.
\end{aligned}$$

This proves (F.20), and thus we proved the first equation in Eq. (F.11). The inequality in Eq. (F.11) is direct since $\exp\{[X_{\succeq x_g} A \log A]/XA\} \le A$. This proves the lemma. $\square$

Proof of Lemma 150

Proof of Lemma 150. We only need to verify that

$$\overline{F}^t := F^{\mathsf{EFCE}}_{\mathsf{bal}}\Big(\eta\sum_{s=1}^{t}\widetilde{M}^s\Big) - F^{\mathsf{EFCE}}_{\mathsf{bal}}\Big(\eta\sum_{s=1}^{t-1}\widetilde{M}^s\Big) \tag{F.21}$$

can be computed via recursive formulas (F.13)-(F.16).

Define

$$G^{\star,t}_{x_ga_g,x_h} := F^{\star}_{x_ga_g,x_h}\left(\eta\sum_{s=1}^{t}\widetilde{M}^s\right) - F^{\star}_{x_ga_g,x_h}\left(\eta\sum_{s=1}^{t-1}\widetilde{M}^s\right).$$

By the definition of $F^{\star}_{x_ga_g,x_h}$ as in Eq. (7.17), we have

$$\begin{aligned}
&G^{\star,t}_{x_ga_g,x_h}\\
=&\frac{1}{\mu^{\star}_{g:h}(x_h,a_h)}\\
&\cdot\log\frac{\sum_{a_h\in\mathcal{A}}\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\times\Big\{F^{\star}_{x_ga_g,x_h}(\mathbf{0})+\sum_{s=1}^{t}\big[-\eta\widetilde{M}^s_{x_ha_h,x_ga_g}+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}G^{\star,s}_{x_ga_g,x_{h+1}}\big]\Big\}\Big\}}{\sum_{a_h\in\mathcal{A}}\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\times\Big\{F^{\star}_{x_ga_g,x_h}(\mathbf{0})+\sum_{s=1}^{t-1}\big[-\eta\widetilde{M}^s_{x_ha_h,x_ga_g}+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}G^{\star,s}_{x_ga_g,x_{h+1}}\big]\Big\}\Big\}}\\
=&\frac{1}{\mu^{\star}_{g:h}(x_h,a_h)}\log\sum_{a_h\in\mathcal{A}}n^t_{x_ga_g,h}(a_h|x_h)\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\times\big[-\eta\widetilde{M}^t_{x_ha_h,x_ga_g}+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}G^{\star,t}_{x_ga_g,x_{h+1}}\big]\Big\},
\end{aligned}$$

where

$$\begin{aligned}
n^t_{x_ga_g,h}(a_h|x_h)\propto_{a_h}\exp\Big\{&\mu^{\star,h}_{g:h}(x_h,a_h)F^{\star}_{x_ga_g,x_h}(\mathbf{0})\\
&+\mu^{\star,h}_{g:h}(x_h,a_h)\sum_{s=1}^{t-1}\Big(-\eta\widetilde{M}^s_{x_ha_h,x_ga_g}+\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)}G^{\star,s}_{x_ga_g,x_{h+1}}\Big)\Big\}.
\end{aligned}$$

Because $\mu^{\star,h}_{g:h}(x_h,a_h)$ and $F^{\star}_{x_ga_g,x_h}(\mathbf{0})$ are independent of a_h for any $(x_h,a_h)\succeq(x_g,a_g)$ (see proof of Lemma 149), we have

$$n^t_{x_ga_g,h}(a_h|x_h)\propto_{a_h}\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\sum_{s=1}^{t-1}\Big(-\eta\widetilde{M}^s_{x_ha_h,x_ga_g}+\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)}G^{\star,s}_{x_ga_g,x_{h+1}}\Big)\Big\}.$$

So $G^{\star,t}_{x_ga_g,x_h}$ and $F^{\star,t}_{x_ga_g,x_h}$ (c.f. Eq. (F.14)) have the same recursive formula, which means that $G^{\star,t}_{x_g a_g, x_h} = F^{\star,t}_{x_g a_g, x_h}$ for any $x_g a_g$ and $x_h \succeq x_g$.

Finally, by the definition of $\overline{F}^t$ as in Eq. (F.21) and the definition of $F^{\mathsf{EFCE}}_{\mathsf{bal}}$ as in Eq. (7.16), we have

$$
\begin{aligned}
&\overline{F}^t \\
&= XA \log \frac{\sum_{g,x_g,a_g} \exp\left\{\frac{1}{XA}\big[-\langle I - E_{\succeq x_g a_g}, \eta \sum_{s=1}^{t} \widetilde{M}^s\rangle + F^{\star}_{x_g a_g, x_g}(\eta \sum_{s=1}^{t} \widetilde{M}^s)\big]\right\}}{\sum_{g,x_g,a_g} \exp\left\{\frac{1}{XA}\big[-\langle I - E_{\succeq x_g a_g}, \eta \sum_{s=1}^{t-1} \widetilde{M}^s\rangle + F^{\star}_{x_g a_g, x_g}(\eta \sum_{s=1}^{t-1} \widetilde{M}^s)\big]\right\}} \\
&= XA \log \frac{\sum_{g,x_g,a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0})\right\} \exp\left\{\frac{1}{XA}\big[\sum_{s=1}^{t} \Big(-\langle I - E_{\succeq x_g a_g}, \eta\ \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\Big)\big]\right\}}{\sum_{g,x_g,a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0})\right\} \exp\left\{\frac{1}{XA}\big[\sum_{s=1}^{t-1} \Big(-\langle I - E_{\succeq x_g a_g}, \eta\ \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\Big)\big]\right\}} \\
&= XA \log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\left\{\frac{1}{XA}[-\eta\langle I - E_{\succeq x_g a_g}, \widetilde{M}^t\rangle + F^{\star,t}_{x_g a_g, x_g}]\right\},
\end{aligned}
$$

where

$$
\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0}) + \frac{1}{XA}\sum_{s=1}^{t-1}\Big(-\eta\langle I - E_{\succeq x_g a_g}, \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\Big)\right\}.
$$

This proves the lemma. □

Bound on $\Delta^t_{x_g a_g}$ via Hessian

The following lemma can be proved similar to Lemma D.11 in Bai et al. [2022b] by calculating the Hessian of $\Delta^t_{x_g a_g}$ with respect to $\widetilde{\ell}^{t, x_g a_g}$. This result is the starting point of both Lemma 151 and Lemma 152.

Lemma 154 (Bound on $\Delta^t_{x_g a_g}$)**.** *We have, almost surely,*

$$
\begin{aligned}
\Delta^t_{x_g a_g} \le 2\eta^2 &\sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \mu^{\star,h''}_{g:h''}(x^t_{h''}, a^t_{h''}) m^t_{x_g a_g, h''+1:h'}(x^t_{h'}, a^t_{h'}) m^t_{x_g a_g, g:h}(x^t_h, a^t_h) \\
&\times \widetilde{\ell}^{t, x_g a_g}_h(x^t_h, a^t_h) \widetilde{\ell}^{t, x_g a_g}_{h'}(x^t_{h'}, a^t_{h'}) \cdot (\mu^t_{x_g a_g})^2 \mathbf{1}\left\{(x^t_{h'}, a^t_{h'}) \succeq (x^t_h, a^t_h) \succeq x_g\right\} \\
\le 2\eta^2 &\sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu^{\star,h''}_{g:h''}(x_{h''}, a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'}, a_{h'}) m^t_{x_g a_g, g:h}(x_h, a_h) (\mu^t_{x_g a_g})^2 \\
&\times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{(\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}))}
\end{aligned}
$$

$$
\times \frac{\mathbf{1}\left\{x_{h'}^t, a_{h'}^t = x_{h'}, a_{h'}\right\}}{(\mu_{1:h'}^t(x_{h'}, a_{h'}) + \gamma(\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'}) + \mu_{x_ga_g}^t m_{x_ga_g,g:h'}^t(x_{h'}, a_{h'})\mathbf{1}\left\{x_{h'} \succeq x_g\right\}))}
\times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\}.
$$

Proof. First, notice that $F_{x_ga_g,x_h}^{\star,t}$ has a similar form as Ξ_1^t in Appendix D.6 of Bai et al. [2022b] with three minor differences:

- $m_{x_ga_g,h}^t$ is used instead of μ_h^t for layer h,
- $\mu_{x_ga_g}^t \widetilde{\ell}_h^{t,x_ga_g}$ is used instead of $\widetilde{\ell}_h^t$ [1] for layer h,
- We terminate the induction argument earlier at (x_g, a_g) instead of the root of the full tree.

Therefore $\Delta_{x_ga_g}^t$ defined as below (cf. (F.17)) is exactly the non-linear part (i.e. remainder term of the first-order Taylor expansion with respect to $\mu_{x_ga_g}^t \widetilde{\ell}^{t,x_ga_g}$ around 0) within $F_{x_ga_g,x_g}^{\star,t}$:

$$
\Delta_{x_ga_g}^t := F_{x_ga_g,x_h}^{\star,t} + \eta\langle m_{x_ga_g}^t \mu_{x_ga_g}^t, \widetilde{\ell}^{t,x_ga_g}\rangle.
$$

Further taking the above differences into account and following exactly the same analysis as in Appendix D.6.1 of Bai et al. [2022b], we get the inequality as claimed. □

Proof of Lemma 151

Proof of Lemma 151. By Lemma 154, for any x_ga_g, we have

$$
\begin{aligned}
&\Delta_{x_ga_g}^t \\
&\le 2\eta^2 \sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_ga_g,h''+1:h'}^t(x_{h'}, a_{h'}) m_{x_ga_g,g:h}^t(x_h, a_h) (\mu_{x_ga_g}^t)^2 \\
&\qquad \times \frac{\mathbf{1}\left\{x_h^t, a_h^t = x_h, a_h\right\}}{(\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_ga_g}^t m_{x_ga_g,g:h}^t(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}))}
\end{aligned}
$$

[1] This is defined in Bai et al. [2022b]. The only difference of $\widetilde{\ell}_h^t$ from $\widetilde{\ell}_h^{t,x_ga_g}$ is that there is no $\gamma\mu_{x_ga_g}^t m_{x_ga_g,g:h}^t(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}$ in the denominator and the indicator function in the numerator.

$$\begin{aligned}
&\times \frac{\mathbf{1}\left\{x_{h'}^t, a_{h'}^t = x_{h'}, a_{h'}\right\}}{\left(\mu_{1:h'}^t(x_{h'}, a_{h'}) + \gamma\left(\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'}) + \mu_{x_ga_g}^t m_{x_ga_g,g:h'}^t(x_{h'}, a_{h'})\mathbf{1}\left\{x_{h'} \succeq x_g\right\}\right)\right)} \\
&\times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\} \\
\le\; & 2\eta^2 \sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_ga_g,h''+1:h'}^t(x_{h'}, a_{h'}) m_{x_ga_g,g:h}^t(x_h, a_h) (\mu_{x_ga_g}^t)^2 \\
&\times \frac{\mathbf{1}\left\{x_h^t, a_h^t = x_h, a_h\right\}}{\gamma \mu_{x_ga_g}^t m_{x_ga_g,g:h}^t(x_h, a_h)} \times \frac{\mathbf{1}\left\{x_{h'}^t, a_{h'}^t = x_{h'}, a_{h'}\right\}}{\gamma \mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'})} \times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\} \\
\le\; & \frac{2\eta^2}{\gamma^2} \sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \frac{\mu_{x_ga_g}^t \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_ga_g,h''+1:h'}^t(x_{h'}, a_{h'})}{\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'})} \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq x_g\right\} \\
\overset{(i)}{\le}\; & \frac{2\eta^2}{\gamma^2} \sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \\
& \frac{\mu_{1:g-1}^t(x_{g-1}, a_{g-1}) \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_ga_g,h''+1:h'}^t(x_{h'}, a_{h'}) \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq x_g\right\}}{\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'})} \\
\overset{(ii)}{\le}\; & \frac{2\eta^2}{\gamma^2} \sum_{g \le h \le h' \le H} \sum_{h''=g}^{h} X_{h'} A \\
\le\; & \frac{2\eta^2}{\gamma^2} H^2 X A.
\end{aligned}$$

Here, (i) uses that

$$\mu_{x_ga_g}^t = \mu_{1:g}^t(x_g, a_g) \le \mu_{1:g-1}^t(x_{g-1}, a_{g-1}),$$

and (ii) uses the property of the balanced policy as in Lemma 37, and observing that

$$\mu_{1:g-1}^t(x_{g-1}, a_{g-1}) \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_ga_g,h''+1:h'}^t(x_{h'}, a_{h'}) \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq x_g\right\}$$

is bounded by some sequence form policy over steps $1 : h'$.

Taking supremum over $x_g a_g$, we get

$$\frac{1}{XA} \sup_{g, x_ga_g} \Delta_{x_ga_g}^t \le \frac{2\eta^2}{\gamma^2} H^2.$$

This proves the lemma. □

Proof of Lemma 152

Proof of Lemma 152. We first upper bound I_t:

$$\begin{aligned}
\mathrm{I}_t \le & 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \lambda^t_{x_g a_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'},a_{h'}) m^t_{x_g a_g, g:h}(x_h,a_h) (\mu^t_{x_g a_g})^2 \\
& \times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{(\mu^t_{1:h}(x_h,a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h,a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h,a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}))} \\
& \times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\}}{(\mu^t_{1:h'}(x_{h'},a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'},a_{h'})\mathbf{1}\left\{x_{h'} \succeq x_g\right\}))} \\
& \times \mathbf{1}\left\{(x_{h'},a_{h'}) \succeq (x_h,a_h) \succeq x_g\right\} \\
\overset{(i)}{\le} & 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'},a_{h'}) \mu^t_{x_g a_g} \mu^t_{1:h}(x_h,a_h) \\
& \times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{(\mu^t_{1:h}(x_h,a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h,a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h,a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}))} \\
& \times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\}}{(\mu^t_{1:h'}(x_{h'},a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'},a_{h'})\mathbf{1}\left\{x_{h'} \succeq x_g\right\}))} \\
& \times \mathbf{1}\left\{(x_{h'},a_{h'}) \succeq (x_h,a_h) \succeq x_g\right\} \\
\le & 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'},a_{h'}) \mu^t_{x_g a_g} \\
& \times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\} \times \mathbf{1}\left\{(x_{h'},a_{h'}) \succeq (x_h,a_h) \succeq x_g\right\}}{(\mu^t_{1:h'}(x_{h'},a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'},a_{h'})\mathbf{1}\left\{x_{h'} \succeq x_g\right\}))} \\
= & 8\eta H \sum_{g\le h'\le H} \sum_{h''=g}^{h'} \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'},a_{h'}) \mu^t_{x_g a_g} \\
& \times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\} \times \mathbf{1}\left\{x_{h'} \succeq x_g\right\}}{\mu^t_{1:h'}(x_{h'},a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'},a_{h'}))} \\
\overset{(ii)}{=} & 8\eta H \sum_{g\le h'\le H} \sum_{h''=g}^{h'} \widetilde{\Delta}^t_{g,h',h''}.
\end{aligned}$$

Here, (i) used the fact that $\lambda^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h,a_h) \mu^t_{x_g a_g} \le \mu^t_{1:h}(x_h,a_h)$ as shown in Eq. (F.6). Moreover, in (ii), we define

$$\widetilde{\Delta}^t_{g,h',h''} = \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'},a_{h'}) \mu^t_{x_g a_g}$$

$$\times \frac{\mathbf{1}\left\{x_{h'}^t, a_{h'}^t = x_{h'}, a_{h'}\right\} \times \mathbf{1}\left\{x_{h'} \succeq x_g\right\}}{\mu_{1:h'}^t(x_{h'}, a_{h'}) + \gamma(\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'}) + \mu_{x_g a_g}^t m_{x_g a_g, g:h'}^t(x_{h'}, a_{h'}))}$$

As a result, the random variable $\widetilde{\Delta}_{g,h',h''}^t$ satisfies the following properties:

[leftmargin=1.5pc]

- $\widetilde{\Delta}_{g,h',h''}^t \le X_{h'}A/\gamma$ almost surely. First,

  $$\widetilde{\Delta}_{g,h,h''}^t \le \frac{1}{\gamma} \sum_{x_{h'},a_{h'}} \frac{\sum_{x_g a_g} \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_g a_g, h''+1:h'}^t(x_{h'}, a_{h'}) \mu_{x_g a_g}^t \mathbf{1}\left\{x_{h'} \succeq x_g\right\}}{\mu_{1:h'}^{\star,h'}(x_{h'}, a_{h'})}.$$

  Notice that (for this fixed g, h'')

  $$\sum_{x_g a_g} \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_g a_g, h''+1:h'}^t(x_{h'}, a_{h'}) \mu_{x_g a_g}^t \mathbf{1}\left\{x_{h'} \succeq x_g\right\} \tag{F.22}$$

  is the sequence-form of a certain policy at $(x_{h'}, a_{h'})$, where the policy is defined as follows: First, take policy $\mu_{1:g}^t$ and arrive at some $x_g \in \mathcal{X}_g$. Let a_g be the action sampled from $\mu_g^t(\cdot|x_g)$. Then, starting from x_g, discard a_g and instead take policy $\mu_{g:h''}^{\star,h''} m_{x_g a_g, h''+1:H}^t$ until the end of the game. One may check that the sequence-form of this policy is indeed given by (F.22). Therefore, we have $\widetilde{\Delta}_{g,h,h''}^t \le X_{h'}A/\gamma$ by the balancing property of $\mu_{1:h'}^{\star,h'}$ (Lemma 37).

- $\mathbb{E}[\widetilde{\Delta}_{g,h',h''}^t|\mathcal{F}_{t-1}] \le 1$: we have

  $$\begin{aligned}&\mathbb{E}[\widetilde{\Delta}_{g,h',h''}^t|\mathcal{F}_{t-1}] \\ &\le \sum_{x_{h'},a_{h'}} \sum_{x_g a_g} \mu_{g:h''}^{\star,h''}(x_{h''}, a_{h''}) m_{x_g a_g, h''+1:h'}^t(x_{h'}, a_{h'}) \mu_{x_g a_g}^t p^t(x_{h'}) \mathbf{1}\left\{x_{h'} \succeq x_g\right\} = 1.\end{aligned}$$

  Above, the last equality used again the fact that (F.22) is the sequence-form of a policy.

- $\mathbb{E}[(\widetilde{\Delta}_{g,h',h''}^t)^2|\mathcal{F}_{t-1}] \le X_{h'}A/\gamma$: note that $\widetilde{\Delta}_{g,h',h''}^t$ is non-negative, so by the almost sure bound that $\widetilde{\Delta}^t_{g,h',h''} \le X_{h'}A/\gamma$, we have

$$\mathbb{E}[(\widetilde{\Delta}^t_{g,h',h''})^2|\mathcal{F}_{t-1}] \le \mathbb{E}[\widetilde{\Delta}^t_{g,h',h''}|\mathcal{F}_{t-1}] \cdot X_{h'}A/\gamma \le X_{h'}A/\gamma.$$

By Freedman's inequality (Lemma 119) and taking the union bound, with probability at least $1 - \delta/(10H^3)$ and some fixed $\lambda \le \gamma/(X_{h'}A)$, we get

$$\sum_{t=1}^{T} \widetilde{\Delta}^t_{g,h,h''} \le T + \frac{\lambda X_{h'}AT}{\gamma} + \frac{4\log(H/\delta)}{\lambda}.$$

Taking $\lambda = \gamma/(X_{h'}A)$, we have

$$\sum_{t=1}^{T} \widetilde{\Delta}^t_{g,h,h''} \le 2T + \frac{4X_{h'}A\log(H/\delta)}{\gamma}.$$

Finally summing up $\widetilde{\Delta}^t_{g,h,h''}$ over g, h, h'' and taking the union bound, we have with probability at least $1 - \delta/10$, we have

$$\sum_{t=1}^{T} \mathrm{I}_t \le 16\eta H^4 T + \frac{32\eta H^3 XA\iota}{\gamma},$$

where $\iota := \log(10H/\delta)$ is the log-factor. This proves the lemma. □

Proof of Lemma 153

Proof. First, recall that the matrix loss estimator is defined as

$$\widetilde{M}^t := \sum_{g,x_g a_g} \mu^t_{g,x_g a_g} \widetilde{\ell}^{t,x_g a_g} e^\top_{x_g a_g}$$

, and the vector loss estimator is computed by

$$\widetilde{\ell}^{t,x_g a_g}_h(x_h, a_h) = \frac{\mathbf{1}\left\{(x^t_h, a^t_h) = (x_h, a_h)\right\}(1 - r^t_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})}.$$

We define a vector $\widetilde{\ell}^t = \{\widetilde{\ell}^t_h(x_h, a_h)\}_{(x_h,a_h)\in\mathcal{X}\times\mathcal{A}} \in \mathbb{R}^{XA}_{\geq 0}$ as

$$\widetilde{\ell}^t_h(x_h, a_h) := \frac{\mathbf{1}\left\{(x^t_h, a^t_h) = (x_h, a_h)\right\}(1 - r^t_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma\mu^{\star,h}_{1:h}(x_h, a_h)}. \tag{F.23}$$

Then we have for any $(t, x_g a_g)$, (x_h, a_h) that

$$\widetilde{\ell}^{t,x_g a_g}_h(x_h, a_h) \leq \widetilde{\ell}^t_h(x_h, a_h).$$

Then $\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle$ can be upper bounded as follows:

$$\begin{aligned}
&\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle \\
&= \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \sum_{h,x_h a_h} \mu^t_{x_h a_h} \widetilde{\ell}^{t,x_g a_g} e^\top_{x_h a_h}\rangle \\
&\leq \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{\ell}^t \sum_{h,x_h,a_h} \mu^t_{x_h a_h} e^\top_{x_h a_h}\rangle \\
&= \langle \phi_{x_g a_g \to m^t_{x_g a_g}}, \widetilde{\ell}^t (\mu^t)^\top\rangle \\
&= \sum_{h=1}^{H} \sum_{h,x_h a_h} (\phi_{x_g a_g \to m^t_{x_g a_g}} \mu^t)_{1:h}(x_h, a_h) \widetilde{\ell}^t_h(x_h, a_h),
\end{aligned}$$

where we have used $\phi_{x_g a_g \to m^t_{x_g a_g}} := I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}$ to denote the EFCE modification triggered at (x_g, a_g) and then playing the policy $m^t_{x_g a_g}$. Also, $\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle \geq 0$ as both matrices have non-negative entries. As a result, we get that

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{g,x_g a_g} \lambda^t_{x_g a_g} \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 \\
&\leq \sum_{t=1}^{T} \sum_{g,x_g a_g} \lambda^t_{x_g a_g} \Big(\langle \phi_{x_g a_g \to m^t_{x_g a_g}} \mu^t, \widetilde{\ell}^t\rangle\Big)^2 \\
&\leq 2\sum_{t=1}^{T} \sum_{g,x_g a_g} \lambda^t_{x_g a_g} \sum_{1\leq h\leq h'\leq H} \sum_{x_h,a_h} \sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)} \\
&\qquad \frac{(\phi_{x_g a_g \to m^t_{x_g a_g}} \mu^t)_{1:h}(x_h, a_h) \mathbf{1}\left\{(x^t_{h'}, a^t_{h'}) = (x_{h'}, a_{h'})\right\} \cdot (1 - r^t_h) \cdot (1 - r^t_{h'})}{(\mu^t_{1:h}(x_h, a_h) + \gamma\mu^{\star,h}_{1:h}(x_h, a_h))(\mu^t_{1:h'}(x_{h'}, a_{h'}) + \gamma\mu^{\star,h'}_{1:h'}(x_{h'}, a_{h'}))}
\end{aligned}$$

$$
\begin{aligned}
&\le 2 \sum_{1\le h\le h'\le H} \sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'})\in \mathcal{C}_{h'}(x_h, a_h)} \sum_{g, x_g a_g} \frac{\lambda^t_{x_g a_g}(\phi_{x_g a_g \to m^t_{x_g a_g}} \mu^t)_{1:h}(x_h, a_h)}{\mu^t_{1:h}(x_h, a_h)} \widetilde{\ell}^t_{h'}(x_{h'}, a_{h'}) \\
&\overset{(i)}{=} 2 \sum_{1\le h\le h'\le H} \sum_{t=1}^{T} \sum_{x_h, a_h} \sum_{(x_{h'}, a_{h'})\in \mathcal{C}_{h'}(x_h, a_h)} \widetilde{\ell}^t_{h'}(x_{h'}, a_{h'}) \\
&\le 2H \sum_{t=1}^{T} \sum_{h', x_{h'}, a_{h'}} \widetilde{\ell}^t_{h'}(x_{h'}, a_{h'}) \\
&\overset{(ii)}{\le} 2H \sum_{t=1}^{T} \sum_{h', x_{h'}, a_{h'}} \underbrace{\ell^t_{h'}(x_{h'}, a_{h'})}_{\le 1} + 2H \sum_{h', x_{h'}, a_{h'}} \frac{\log(10XA/\delta)}{\gamma \mu^{\star, h'}_{1:h'}(x_{h'}, a_{h'})} \\
&\overset{(iii)}{\le} 2HXAT + 2H \sum_{h', x_{h'}, a_{h'}} \iota \cdot X_{h'} A/\gamma \\
&\le 2HXAT + 2HX^2A^2\iota/\gamma.
\end{aligned}
$$

Above, (i) uses that μ^t is the solution of the fixed point equation $\mu = \sum_{x_g a_g} \lambda^t_{x_g, a_g}(I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g})\mu$; (ii) is by [Bai et al., 2022b, Corollary D.6] for each $(h', x_{h'}, a_{h'})$ with probability $1-\delta/(10XA)$ and a union bound; (iii) uses $\mu^{\star, h'}_{1:h'}(x_{h'}, a_{h'}) \ge 1/(X_{h'}A)$ by Corollary 122. Therefore, we have with probability at least $1 - \delta/10$ that

$$
\begin{aligned}
\sum_{t=1}^{T} \mathrm{II}_t &= \frac{e\eta}{XA} \cdot \sum_{t=1}^{T} \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t \rangle^2 \\
&\le \frac{e\eta}{XA}\big(2HXAT + 2HX^2A^2\iota/\gamma\big) \le 6\eta HT + 6\eta HXA\iota/\gamma.
\end{aligned}
$$

This proves the lemma. □

F.3.3 Bound on two bias terms

Proof of Lemma 147. First, recall that the matrix loss estimator gives

$$
\widetilde{M}^t = \sum_{g, x_g, a_g} \mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g} e^\top_{x_g a_g}
$$

and the vector loss estimator is computed by

$$\widetilde{\ell}_h^{t,x_g a_g}(x_h, a_h) = \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\}(1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})}.$$

Then we decompose BIAS_1 as

$$\begin{aligned}\mathrm{BIAS}_1 &= \sum_{t=1}^T \langle \phi^t, \ell^t(\mu^t)^\top - \widetilde{M}^t\rangle \\ &= \underbrace{\sum_{t=1}^T \langle \phi^t, \ell^t(\mu^t)^\top - \mathbb{E}\Big[\widetilde{M}^t|\mathcal{F}_{t-1}\Big]\rangle}_{(A)} + \underbrace{\sum_{t=1}^T \langle \phi^t, \mathbb{E}\Big[\widetilde{M}^t|\mathcal{F}_{t-1}\Big] - \widetilde{M}^t\rangle}_{(B)}.\end{aligned}$$

We first the second term (B) by Azuma-Hoeffding inequality. Recall the definition of $\widetilde{\ell}^t$ in (F.23). We immediately have $\widetilde{\ell}^{t,x_g a_g} \le \widetilde{\ell}^t$ pointwisely, so we can upper bound $\langle \phi^t, \widetilde{M}^t\rangle$ by

$$\langle \phi^t, \widetilde{M}^t\rangle \le \langle \phi^t, \sum_{g,x_g,a_g} \mu_{x_g a_g}^t \widetilde{\ell}^t e_{x_g a_g}^\top\rangle = \langle \phi^t \mu^t, \widetilde{\ell}^t\rangle = \langle \mu^t, \widetilde{\ell}^t\rangle,$$

where the last equality comes from fixed point equation $\mu^t = \phi^t \mu^t$. Then we have

$$\begin{aligned}&\langle \phi^t, \widetilde{M}^t\rangle \le \langle \mu^t, \widetilde{\ell}^t\rangle \\ &= \sum_{h=1}^H \sum_{x_h,a_h} \mu_{1:h}^t(x_h, a_h) \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star,h}(x_h, a_h)} \\ &\le \sum_{h=1}^H \sum_{x_h,a_h} \mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\} = \sum_{h=1}^H 1 = H.\end{aligned}$$

As a consequence, by Azuma-Hoeffding inequality, with probability at least $1 - \delta/10$, we have

$$\sum_{t=1}^T \langle \phi^t, \mathbb{E}\Big[\widetilde{M}^t|\mathcal{F}_{t-1}\Big] - \widetilde{M}^t\rangle \le H\sqrt{2T\log(10/\delta)} \le H\sqrt{2T\iota}.$$

Then we turn to bound the first term (A). Denote $\ell^{t,x_g a_g} = \mathbb{E}\Big[\widetilde{\ell}^{t,x_g a_g}|\mathcal{F}_{t-1}\Big]$ and plug in the definition of $\widetilde{\ell}^{t,x_g a_g}$, we get

$$\begin{aligned}
&\langle \phi^t, \ell^t(\mu^t)^\top - \mathbb{E}\Big[\widetilde{M}^t | \mathcal{F}_{t-1}\Big]\rangle \\
=&\langle \phi^t, \ell^t(\mu^t)^\top\rangle - \sum_{g,x_g,a_g} \langle \phi^t, \mu^t_{x_g a_g} \ell^{t,x_g a_g} e^\top_{x_g a_g}\rangle \\
=& \sum_{g,x_g,a_g} \langle \phi^t e_{x_g a_g} \mu^t_{x_g a_g}, \ell^t - \ell^{t,x_g a_g}\rangle.
\end{aligned}$$

Note that by the definition of the loss estimator as in Eq. (5.10), we have

$$\ell^t_h(x_h, a_h) = p^t(x_h)[1 - \overline{R}^t_h(x_h, a_h)] \leq p^t(x_h),$$

where we recall the definition of $p^t(x_h)$ in (F.7).

Moreover, the $\ell^{t,x_g a_g}(x_h, a_h)$ is related to $\ell^t_h(x_h, a_h)$ by a rescaling

$$\ell^{t,x_g a_g}(x_h, a_h) = \frac{\mu^t_{1:h}(x_h, a_h)\ell^t_h(x_h, a_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})}.$$

So we get

$$\begin{aligned}
&\langle \phi^t, \ell^t(\mu^t)^\top - \mathbb{E}\Big[\widetilde{M}^t | \mathcal{F}_{t-1}\Big]\rangle \\
=& \sum_{g,x_g,a_g} \sum_{h,x_h,a_h} \mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \\
&\quad \times \frac{\gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})\ell^t_h(x_h, a_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})} \\
\leq& \gamma \sum_{g,x_g,a_g} \sum_{h,x_h,a_h} \frac{\mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \mu^{\star,h}_{1:h}(x_h, a_h) p^t(x_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})} \\
&+ \gamma \sum_{g,x_g,a_g} \sum_{h,x_h,a_h} \frac{\mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\} p^t(x_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\left\{x_h \succeq x_g\right\})}.
\end{aligned}$$

The first term admits an upper bound

$$\gamma \sum_{h,x_h,a_h} \sum_{g,x_g,a_g} \frac{\mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h)}{\mu^t_{1:h}(x_h, a_h)} \mu^{\star,h}_{1:h}(x_h, a_h) p^t(x_h)$$

$$\overset{(i)}{=} \gamma \sum_{h,x_h,a_h} \mu^{\star,h}_{1:h}(x_h,a_h)p^t(x_h) \overset{(ii)}{=} \gamma H.$$

Here, (i) uses $\mu^t = \phi^t \mu^t$ and (ii) uses Eq. (F.8).

The second term can be upper bounded by

$$\begin{aligned}
&\gamma \sum_{h,x_h,a_h} \sum_{g,x_g,a_g,x_h\succeq x_g} \frac{\mu^t_{x_g a_g}(\phi^t e_{x_g a_g})_{1:h}(x_h,a_h)\mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)p^t(x_h)}{\mu^t_{1:h}(x_h,a_h)} \\
\le\ & \gamma \sum_{h,x_h,a_h} \left(\sum_{g,x_g,a_g,x_h\succeq x_g} \frac{\mu^t_{x_g a_g}(\phi^t e_{x_g a_g})_{1:h}(x_h,a_h)}{\mu^t_{1:h}(x_h,a_h)} \right) \cdot \left(\sum_{g,x_g,a_g,x_h\succeq x_g} \mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)p^t(x_h) \right) \\
\overset{(i)}{\le}\ & \gamma \sum_{h,x_h,a_h} \sum_{g,x_g,a_g,x_h\succeq x_g} \mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)p^t(x_h) \\
=\ & \gamma \sum_{h,x_h,a_h} \sum_{g=1}^{h} \sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)\mathbf{1}\left\{x_h \succeq x_g\right\} \cdot p^t(x_h) \\
=\ & \gamma \sum_{1\le g\le h\le H} \sum_{x_h,a_h} \sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)\mathbf{1}\left\{x_h \succeq x_g\right\} \cdot p^t(x_h) \\
\overset{(ii)}{\le}\ & \gamma H^2.
\end{aligned}$$

Here, the inequality in (i) also uses $\mu^t = \phi^t \mu^t$; (ii) used the fact for any fixed g, $\sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g,g:h}(x_h,a_h)\mathbf{1}\left\{x_h \succeq x_g\right\}$ is the sequence-form of a policy, similar as (F.22).

Taking summation over $t = 1, 2, \cdots, T$, we have

$$(A) \le 2\gamma H^2 T.$$

Combined with the bound on (B), we have

$$\mathrm{BIAS}_1 \le 2\gamma H^2 T + 2H\sqrt{T\iota}.$$

This completes the proof of this lemma. □

Proof of Lemma 148. Recall the definition of $\widetilde{\ell}^t$ in (F.23). We have $\widetilde{\ell}^{t,x_g a_g} \le \widetilde{\ell}^t$ pointwisely, so we have

$$\begin{aligned}\langle \phi^\star, \widetilde{M}^t - \ell^t(\mu^t)^\top \rangle &= \langle \phi^\star, \sum_{g, x_g a_g} \widetilde{\ell}^{t, x_g a_g} \mu^t_{x_g a_g} e^\top_{x_g a_g} - \ell^t(\mu^t)^\top \rangle \\ &\le \langle \phi^\star, \widetilde{\ell}^t(\mu^t)^\top - \ell^t(\mu^t)^\top \rangle = \langle \phi^\star \mu^t, \widetilde{\ell}^t - \ell^t \rangle.\end{aligned}$$

Then we can get that with probability at least $1 - \delta/3$

$$\begin{aligned}\mathrm{BIAS}_2 &\le \max_{\phi^\star \in \Phi^{\mathsf{EFCE}}} \sum_{t=1}^T \left\langle \phi^\star \mu^t, \widetilde{\ell}^t - \ell^t \right\rangle \\ &= \max_{\phi^\star \in \Phi^{\mathsf{EFCE}}} \sum_{t=1}^T \sum_{h=1}^H \sum_{x_h, a_h} (\phi^\star \mu^t)_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\ &= \max_{\phi^\star \in \Phi^{\mathsf{EFCE}}} \sum_{t=1}^T \sum_{h=1}^H \sum_{x_h, a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\gamma \mu^{\star,h}_{1:h}(x_h, a_h)} \gamma \mu^{\star,h}_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\ &= \max_{\phi^\star \in \Phi^{\mathsf{EFCE}}} \sum_{h=1}^H \sum_{x_h, a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\gamma \mu^{\star,h}_{1:h}(x_h, a_h)} \sum_{t=1}^T \gamma \mu^{\star,h}_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\ &\overset{(i)}{\le} \frac{\log(3XA/\delta)}{\gamma} \max_{\phi^\star \in \Phi^{\mathsf{EFCE}}} \sum_{h=1}^H \sum_{x_h, a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \\ &\overset{(ii)}{=} \frac{\iota}{\gamma} \sum_{h=1}^H X_h A = XA\iota/\gamma,\end{aligned}$$

where (i) is a high probability bound by applying Corollary D.6 in Bai et al. [2022b] for each (x_h, a_h) pair and taking union bound, and (ii) is by the balancing property of $\mu^{\star,h}$. This proves the lemma. □

F.4 Equivalence between Vertex MWU and OMD

In this section we prove Theorem 62. Our proof is based on Algorithm 25, which is just (the efficient implementation of) the standard OMD algorithm with dilated entropy regularizer in FTRL form [Kroer et al., 2015]. Indeed, Lemma 157 show that its output policy $\{\mu^t\}_{t\ge1}$ is the same as (7.31). Then, Lemma 155 & 156 show that its output policy $\{\mu^t\}_{t\ge1}$ is the same as (7.30). These together imply the equivalence

Algorithm 25 OMD (FTRL form)

```
Require: Learning rate $\eta > 0$.
for $t = 1, 2, \dots, T$ do
    Compute $\mu_h^t(a_h|x_h)$ and $F_{x_h}^t$ in the bottom-up order over $x_h \in \mathcal{X}$:
        $\mu_h^t(a_h|x_h) \propto_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} \ell_h^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}^t \Big\}$,   (F.24)
        $F_{x_h}^t = \log \sum_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} \ell_h^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}^t \Big\}$.   (F.25)
    Receive loss $\ell^t = \{\ell_h^t(x_h, a_h)\}_{(x_h, a_h) \in \mathcal{X} \times \mathcal{A}} \in \mathbb{R}_{\geq 0}^{XA}$.
```

of (7.31) and (7.30), thereby proving Theorem 62.

The rest of this subsection is devoted to stating and proving Lemma 155-157.

Remark on optimistic algorithms As pointed out in Farina et al. [2022b], Theorem 62 does not depend on the concrete values of $\{\ell^t\}_{t \geq 1}$. As a result, the equivalence also holds for the optimistic version of the algorithms (where the algorithms are fed with loss functions $\{2\ell^t - \ell^{t-1}\}_{t \geq 1}$, with $\ell^0 := 0$) which achieves an faster $\mathcal{O}(\mathrm{poly}(\log T))$ regret. In words: The Kernelized OMWU algorithm of Farina et al. [2022b] is equivalent to an Optimistic OMD algorithm with the dilated KL distance.

Lemma 155 (Conversion to log-partition function)**.** *Define the log-partition function* $F^{\mathcal{V}} : \mathbb{R}^{XA} \to \mathbb{R}$

$$F^{\mathcal{V}}(\ell) := \log \textstyle\sum_{v \in \mathcal{V}} \exp\{-\langle v, \ell \rangle\}. \tag{F.26}$$

Then update (7.30) *has a closed-form update for all* $t \geq 1$*:*

$$\mu^t = -\nabla F^{\mathcal{V}}\big(\eta \textstyle\sum_{s=1}^{t-1} \ell^s\big) = -\frac{\sum_{v \in \mathcal{V}} \exp\left\{-\eta\left\langle v, \sum_{s=1}^{t-1} \ell^s \right\rangle\right\} v}{\sum_{v \in \mathcal{V}} \exp\left\{-\eta\left\langle v, \sum_{s=1}^{t-1} \ell^s \right\rangle\right\}}. \tag{F.27}$$

Proof. By (7.30),

$$\mu^t = \sum_v p_v^t v = \frac{\sum_\phi \exp\{-\eta \langle v, \sum_{s=1}^{t-1} \ell^s \rangle\} v}{\sum_v \exp\{-\eta \langle v, \sum_{s=1}^{t-1} \ell^s \rangle\}} = -\nabla F^{\mathcal{V}}\left(\eta \sum_{s=1}^{t-1} \ell^s\right).$$

□

Lemma 156 (Recursive expression of $F^{\mathcal{V}}$ and $\nabla F^{\mathcal{V}}$)**.** *For any loss matrix* $\ell \in \mathbb{R}^{XA}$*, the* log-partition function *can be written as* $F^{\mathcal{V}}(\ell) = F_{\phi}(\ell)$ *where* $F_{x_h}(\ell) := \log \sum_{v \in \mathcal{V}^{x_h}} \exp\{-\langle v, \ell \rangle\}$ *can be computed recurrently by* $F_{x_{H+1}}(\cdot) = 0$ *and*

$$F_{x_h}(\ell) := \log \textstyle\sum_{a_h} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}(\ell) \Big\}. \tag{F.28}$$

Furthermore, define a (sequence form) policy μ *by*

$$\mu(a_h | x_h) \propto_{a_h} \exp\Big\{ - \ell_h(x_h, a_h) + \textstyle\sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}(\ell) \Big\}, \tag{F.29}$$

then we have

$$-\nabla F^{\mathcal{V}}(\ell) = \mu. \tag{F.30}$$

Proof. We first show (F.28). Using the structure of $\mathcal{V}^{x_h}$,

$$\begin{aligned} F_{x_h}(\ell) &= \log \sum_{v \in \mathcal{V}^{x_h}} \exp\{-\langle v, \ell \rangle\} \\ &= \log \sum_{a_h \in \mathcal{A}_{x_h}} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \sum_{v \in \mathcal{V}^{x_{h+1}}} \exp\{-\langle v, \ell \rangle\} \Big\}. \\ &= \log \sum_{a_h \in \mathcal{A}_{x_h}} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell) \Big\}. \end{aligned}$$

Next we show (F.30). Taking the gradient,

$$\begin{aligned} & - \nabla F_{x_h}(\ell) \\ = & \frac{\sum_{a_h \in \mathcal{A}_{x_h}} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell) \Big\} [e_{x_h a_h} - \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \nabla F_{x_{h+1}}(\ell)]}{\sum_{a_h \in \mathcal{A}_{x_h}} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell) \Big\}} \\ = & \sum_{a_h \in \mathcal{A}_{x_h}} \mu_h(a_h | x_h) [e_{x_h a_h} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} (-\nabla F_{x_{h+1}})(\ell)]. \end{aligned}$$

For any x_h, repeat the above process along the treeplex, the contribution of the production of $\mu(\cdot|\cdot)$ will be the sequence form. As a result, $-\nabla F^{\mathcal{V}}(\ell) = \mu$, which completes the proof. $\square$

Lemma 157. *The policy μ^t in Algorithm 25 is the optimizer of the optimization problem*

$$\arg\min_{\mu\in\Pi^{x_h}} \left[\eta \left\langle \mu, \textstyle\sum_{s=1}^{t-1} \ell^s \right\rangle + H_{x_h}(\mu)\right]$$

for all x_h simultaneously. Furthermore,

$$\min_{\mu\in\Pi^{x_h}} \left[\eta \left\langle \mu, \textstyle\sum_{s=1}^{t-1} \ell^s \right\rangle + H_{x_h}(\mu)\right] = -F_{x_h}\left(\textstyle\sum_{s=1}^{t-1} \ell^s\right).$$

The result is known in the literature (e.g. Kroer et al. [2015]) and its proof is similar to Appendix B of Kozuno et al. [2021], which focuses on the special case when the loss function is the bandit-based loss estimator (7.15):

$$\ell_h^t(x_h, a_h) := \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\}(1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma}.$$

For completeness, we include a proof for generic loss here.

Proof. We prove by induction for $h = H, \cdots, 1$. For $h = H$, since there is no further decisions to be make, this is just a linear optimization problem with entropy regularizer on simplex. As a result, $\mu_H(a_H|x_H) \propto_{a_H} \exp\{-\ell_H(x_H, a_H)\}$ as desired and the minimum is $-\log\sum_{a_H} \exp\Big\{ - \ell_H(x_H, a_H)\Big\} = -F_{x_H}(\ell)$.

If the claim holds for levels after $h+1$, consider the h-th level. Plug in the optimizer after the $h+1$-th level, the optimization problem in the sub-tree rooted at x_h becomes

$$\arg\min_{\mu\in\Pi^{x_h}} \left[\sum_{a_h} \mu_h(a_h|x_h)(\ell_h(x_h, a_h) - \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell))\right],$$

which is again a linear optimization problem with entropy regularizer on simplex. As a result, $\mu_h(a_h|x_h) \propto_{a_h} \exp\{-\ell_h(x_h, a_h) + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell))\}$ as desired and the minimum is $-\log\sum_{a_h} \exp\Big\{ - \ell_h(x_h, a_h) + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F_{x_{h+1}}(\ell))\Big\} = -F_{x_h}(\ell)$. $\square$

Bibliography

A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646. PMLR, 2014.

A. Agarwal, A. Krishnamurthy, J. Langford, H. Luo, et al. Open problem: First-order regret bounds for contextual bandits. In *Conference on Learning Theory*, pages 4–7. PMLR, 2017.

I. Anagnostides, C. Daskalakis, G. Farina, M. Fishelson, N. Golowich, and T. Sandholm. Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2022a.

I. Anagnostides, G. Farina, C. Kroer, C.-W. Lee, H. Luo, and T. Sandholm. Uncoupled learning dynamics with $O(\log T)$ swap regret in multiplayer games. 35: 3292–3304, 2022b.

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017.

Y. Bai and C. Jin. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*, pages 551–560. PMLR, 2020.

Y. Bai, C. Jin, and T. Yu. Near-optimal reinforcement learning with self-play. In *Advances in Neural Information Processing Systems*, 2020.

Y. Bai, C. Jin, S. Mei, Z. Song, and T. Yu. Efficient Φ-regret minimization in extensive-form games via online mirror descent. *Advances in Neural Information Processing Systems*, 2022a.

Y. Bai, C. Jin, S. Mei, and T. Yu. Near-optimal learning of extensive-form games with imperfect information. In *International Conference on Machine Learning*, pages 1337–1382. PMLR, 2022b.

B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2020.

K. Berg and T. Sandholm. Exclusion method for finding Nash equilibrium in multiplayer games. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1417–1418, 2016.

D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 1956.

A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

R. I. Brafman and M. Tennenholtz. R-max: a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3 (Oct):213–231, 2002.

M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

N. Brown and T. Sandholm. Regret transfer and parameter optimization. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

N. Brown and T. Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 365 (6456):885–890, 2019.

N. Burch, M. Moravcik, and M. Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019.

Y. Cai, A. Oikonomou, and W. Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *Advances in Neural Information Processing Systems*, 2022.

A. Celli, S. Coniglio, and N. Gatti. Computing optimal ex ante correlated equilibria in two-player sequential games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 909–917, 2019a.

A. Celli, A. Marchesi, T. Bianchi, and N. Gatti. Learning to correlate in multi-player general-sum sequential games. *Advances in Neural Information Processing Systems*, 32, 2019b.

A. Celli, A. Marchesi, G. Farina, and N. Gatti. No-regret learning dynamics for extensive-form correlated equilibrium. *Advances in Neural Information Processing Systems*, 33, 2020.

S. Cen, Y. Chi, S. S. Du, and L. Xiao. Faster last-iterate convergence of policy optimization in zero-sum Markov games. 2023.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.* Cambridge university press, 2006.

Q. Cui, Z. Xiong, M. Fazel, and S. S. Du. Learning in congestion games with bandit feedback. In *Advances in Neural Information Processing Systems*, 2022.

C. Dann and E. Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

C. Dann, T. Lattimore, and E. Brunskill. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723, 2017.

C. Daskalakis. On the complexity of approximating a Nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):23, 2013.

C. Daskalakis and I. Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *10th Innovations in Theoretical Computer Science (ITCS) conference, ITCS 2019*, 2019.

C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

C. Daskalakis, D. J. Foster, and N. Golowich. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33:5527–5540, 2020.

C. Daskalakis, M. Fishelson, and N. Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34, 2021.

C. Daskalakis, N. Golowich, and K. Zhang. The complexity of Markov equilibrium in stochastic games. *arXiv preprint arXiv:2204.03991*, 2022.

S. Du, A. Krishnamurthy, N. Jiang, A. Agarwal, M. Dudik, and J. Langford. Provably efficient RL with Rich Observations via Latent State Decoding. In *International Conference on Machine Learning*, pages 1665–1674, 2019.

S. Du, S. Kakade, J. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in RL. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

M. Dudík and G. J. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 151–160, 2009.

L. Erez, T. Lancewicki, U. Sherman, T. Koren, and Y. Mansour. Regret minimization and convergence to equilibria in general-sum Markov games. *arXiv preprint arXiv:2207.14211*, 2022.

G. Farina and T. Sandholm. Model-free online learning in unknown sequential decision making problems and games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5381–5390, 2021.

G. Farina, C. K. Ling, F. Fang, and T. Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Advances in Neural Information Processing Systems*, 32, 2019.

G. Farina, T. Bianchi, and T. Sandholm. Coarse correlation in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1934–1941, 2020a.

G. Farina, C. Kroer, and T. Sandholm. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning*, pages 3018–3028. PMLR, 2020b.

G. Farina, C. Kroer, and T. Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5363–5371, 2021a.

G. Farina, R. Schmucker, and T. Sandholm. Bandit linear optimization for sequential decision making and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5372–5380, 2021b.

G. Farina, A. Celli, A. Marchesi, and N. Gatti. Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *Journal of the ACM*, 69(6): 1–41, 2022a.

G. Farina, C.-W. Lee, H. Luo, and C. Kroer. Kernelized multiplicative weights for 0/1-polyhedral games: Bridging the gap between learning in extensive-form and normal-form games. In *International Conference on Machine Learning*, pages 6337–6357. PMLR, 2022b.

C. Fiegel, P. Ménard, T. Kozuno, R. Munos, V. Perchet, and M. Valko. Adapting to game trees in zero-sum imperfect information games. *arXiv preprint arXiv:2212.12567*, 2022.

J. Filar and K. Vrieze. *Competitive Markov decision processes.* Springer Science & Business Media, 2012.

D. J. Foster, S. M. Kakade, J. Qian, and A. Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

D. P. Foster and R. V. Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998.

N. Golowich, S. Pattathil, and C. Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. *Advances in neural information processing systems*, 33:20766–20778, 2020a.

N. Golowich, S. Pattathil, C. Daskalakis, and A. Ozdaglar. Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems. In *Conference on Learning Theory*, pages 1758–1784. PMLR, 2020b.

G. J. Gordon, A. Greenwald, and C. Marks. No-regret learning in convex games. In *Proceedings of the 25th international conference on Machine learning*, pages 360–367, 2008.

A. Greenwald and A. Jafari. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Learning theory and kernel machines*, pages 2–12. Springer, 2003.

T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

J. Heinrich, M. Lanctot, and D. Silver. Fictitious self-play in extensive-form games. In *International conference on machine learning*, pages 805–813. PMLR, 2015.

S. Hoda, A. Gilpin, J. Pena, and T. Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2): 494–512, 2010.

J. Hu and M. P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

W. Huang and B. von Stengel. Computing an extensive-form correlated equilibrium in polynomial time. In *International Workshop on Internet and Network Economics*, pages 506–513. Springer, 2008.

S. Iqbal and F. Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970. PMLR, 2019.

S. K. Jakobsen, T. B. Sørensen, and V. Conitzer. Timeability of extensive-form games. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 191–199, 2016.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Z. Jia, L. F. Yang, and M. Wang. Feature-based Q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*, 2019.

N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire. Contextual decision processes with low Bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4868–4878, 2018.

C. Jin, T. Jin, H. Luo, S. Sra, and T. Yu. Learning adversarial Markov decision processes with bandit feedback and unknown transition. In *International Conference on Machine Learning*, pages 4860–4869. PMLR, 2020a.

C. Jin, A. Krishnamurthy, M. Simchowitz, and T. Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020b.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020c.

C. Jin, Q. Liu, and S. Miryoosefi. Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 2021a.

C. Jin, Q. Liu, Y. Wang, and T. Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent RL. *arXiv preprint arXiv:2110.14555*, 2021b.

C. Jin, Q. Liu, and T. Yu. The power of exploiter: Provable multi-agent RL in large state spaces. In *International Conference on Machine Learning*, pages 10251–10279. PMLR, 2022.

M. Johanson. Measuring the size of large no-limit poker games. *arXiv preprint arXiv:1302.7008*, 2013.

M. Johanson, N. Bard, M. Lanctot, R. G. Gibson, and M. Bowling. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *AAMAS*, pages 837–846. Citeseer, 2012.

D. Kane, S. Liu, S. Lovett, and G. Mahajan. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pages 1282–1302. PMLR, 2022.

H. Kao, C.-Y. Wei, and V. Subramanian. Decentralized cooperative reinforcement learning with hierarchical information structure. In *International Conference on Algorithmic Learning Theory*, pages 573–605. PMLR, 2022.

M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

S. Khot and A. K. Ponnuswami. Minimizing wide range regret with time selection functions. In *COLT*, pages 81–86, 2008.

D. Koller and N. Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and economic behavior*, 4(4):528–552, 1992.

D. Koller, N. Megiddo, and B. Von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and economic behavior*, 14(2):247–259, 1996.

V. Kova**v**rík, M. Schmid, N. Burch, M. Bowling, and V. Lisỳ. Rethinking formal models of partially observable multiagent decision making. *Artificial Intelligence*, 303:103645, 2022.

T. Kozuno, P. Ménard, R. Munos, and M. Valko. Model-free learning for two-player zero-sum partially observable Markov games with perfect recall. In *Advances in Neural Information Processing Systems*, 2021.

C. Kroer, K. Waugh, F. Kilinç-Karzan, and T. Sandholm. Faster first-order methods for extensive-form game solving. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 817–834, 2015.

C. Kroer, G. Farina, and T. Sandholm. Solving large sequential games with the excessive gap technique. *Advances in neural information processing systems*, 31, 2018.

H. W. Kuhn. Extensive games and the problem of information. In *Contributions to the Theory of Games (AM-28), Volume II*, pages 193–216. Princeton University Press, 1953.

M. Lanctot, K. Waugh, M. Zinkevich, and M. H. Bowling. Monte Carlo sampling for regret minimization in extensive games. In *NIPS*, pages 1078–1086, 2009.

T. Lattimore and C. Szepesvári. *Bandit algorithms.* Cambridge University Press, 2020.

C.-W. Lee, C. Kroer, and H. Luo. Last-iterate convergence in extensive-form games. *Advances in Neural Information Processing Systems*, 34:14293–14305, 2021.

M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

M. L. Littman. Friend-or-foe Q-learning in general-sum games. In *ICML*, volume 1, pages 322–328, 2001.

M. Liu, A. Ozdaglar, T. Yu, and K. Zhang. The power of regularization in solving extensive-form games. 2023.

Q. Liu, T. Yu, Y. Bai, and C. Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021.

Q. Liu, Y. Wang, and C. Jin. Learning Markov games with adversarial opponents: Efficient algorithms and fundamental limits. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14036–14053. PMLR, 17–23 Jul 2022.

R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

W. Mao and T. Başar. Provably efficient reinforcement learning in decentralized general-sum markov games. *Dynamic Games and Applications*, 13(1):165–186, 2023.

W. Mao, L. Yang, K. Zhang, and T. Basar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049. PMLR, 2022.

I. Milchtaich. Congestion games with player-specific payoff functions. *Games and economic behavior*, 13(1):111–124, 1996.

D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14 (1):124–143, 1996.

M. Moravv**v**cík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

D. Morrill, R. D'Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald. Efficient deviation types and learning for hindsight rationality in extensive-form games. In *International Conference on Machine Learning*, pages 7818–7828. PMLR, 2021.

E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden Markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375, 2005.

R. Munos, J. Perolat, J.-B. Lespiau, M. Rowland, B. De Vylder, M. Lanctot, F. Timbers, D. Hennes, S. Omidshafiei, A. Gruslys, et al. Fast computation of Nash equilibria in imperfect information games. In *International Conference on Machine Learning*, pages 7119–7129. PMLR, 2020.

J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3168–3176, 2015.

OpenAI. Openai five. `https://blog.openai.com/openai-five/`, 2018.

I. Osband and B. Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

I. Osband, B. Van Roy, and Z. Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386. PMLR, 2016.

M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.

G. Radanovic, R. Devidze, D. Parkes, and A. Singla. Learning to collaborate in Markov decision processes. In *International Conference on Machine Learning*, pages 5261–5270, 2019.

T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

M. Sayin, K. Zhang, D. Leslie, T. Basar, and A. Ozdaglar. Decentralized Q-learning in zero-sum Markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

M. Schmid, N. Burch, M. Lanctot, M. Moravcik, R. Kadlec, and M. Bowling. Variance reduction in Monte Carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2157–2164, 2019.

M. Schmid, M. Moravcik, N. Burch, R. Kadlec, J. Davidson, K. Waugh, N. Bard, F. Timbers, M. Lanctot, Z. Holland, et al. Player of games. *arXiv preprint arXiv:2112.03178*, 2021.

S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

L. S. Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39 (10):1095–1100, 1953.

A. Sidford, M. Wang, L. Yang, and Y. Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pages 2992–3002. PMLR, 2020.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of Go without human knowledge. *nature*, 550(7676):354–359, 2017.

K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

Z. Song, S. Mei, and Y. Bai. When can we learn general-sum markov games with a large number of players sample-efficiently? In *International Conference on Learning Representations*, 2022a.

Z. Song, S. Mei, and Y. Bai. Sample-efficient learning of correlated equilibria in extensive-form games. In *Advances in Neural Information Processing Systems*, 2022b.

G. Stoltz and G. Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1):125–159, 2005.

G. Stoltz and G. Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59(1):187–208, 2007.

A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, pages 881–888, 2006.

P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.

O. Tammelin. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*, 2014.

Y. Tian, Q. Gong, and Y. Jiang. Joint policy search for multi-agent collaboration with imperfect information. *Advances in Neural Information Processing Systems*, 33:19931–19942, 2020.

Y. Tian, Y. Wang, T. Yu, and S. Sra. Online learning in unknown Markov games. In *International Conference on Machine Learning*, pages 10279–10288. PMLR, 2021.

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

B. Von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

B. Von Stengel and F. Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

R. Wang, S. S. Du, L. Yang, and R. R. Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

Y. Wang, Q. Liu, Y. Bai, and C. Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent RL with function approximation. *arXiv preprint arXiv:2302.06606*, 2023.

C.-Y. Wei. Analysis of UCSG in the finite-horizon setting. *Personal Communication.*, 2021.

C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems*, pages 4987–4997, 2017.

C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations*, 2021a.

C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive Markov games. In *Conference on learning theory*, pages 4259–4299. PMLR, 2021b.

Q. Xie, Y. Chen, Z. Wang, and Z. Yang. Learning zero-sum simultaneous-move Markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, pages 3674–3682. PMLR, 2020.

Y. A. Yadkori, P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems*, pages 2508–2516, 2013.

T. Yu, Y. Tian, J. Zhang, and S. Sra. Provably efficient algorithms for multi-objective competitive RL. In *International Conference on Machine Learning*, pages 12167–12176. PMLR, 2021.

A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill. Learning near optimal policies with low inherent Bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

W. Zhan, J. D. Lee, and Z. Yang. Decentralized optimistic hyperpolicy mirror descent: Provably no-regret learning in Markov games. *International Conference on Learning Representations*, 2023.

B. H. Zhang and T. Sandholm. Finding and certifying (near-) optimal strategies in black-box extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5779–5788, 2021.

K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018.

K. Zhang, S. M. Kakade, T. Başar, and L. F. Yang. Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461*, 2020a.

X. Zhang, Y. Ma, and A. Singla. Task-agnostic exploration in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:11734–11743, 2020b.

Z. Zhang, Y. Zhou, and X. Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020c.

Y. Zhou, J. Li, and J. Zhu. Posterior sampling for multi-agent reinforcement learning: solving extensive games with imperfect information. In *International Conference on Learning Representations*, 2019.

Y. Zhou, T. Ren, J. Li, D. Yan, and J. Zhu. Lazy-CFR: fast and near-optimal regret minimization for extensive games with imperfect information. In *International Conference on Learning Representations*, 2020.

M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20:1729–1736, 2007.