Enhancing Learning Algorithms via Sublinear-Time Methods

by

Arsen Vasilyan

B.S., Massachusetts Institute of Technology (2019)
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

May 2024

Authored by Arsen Vasilyan
Department of Electrical Engineering and Computer Science
March 20, 2024

Certified by Jonathan Kelner
Professor of Applied Mathematics
Thesis Supervisor

Certified by Ronitt Rubinfeld
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

Enhancing Learning Algorithms via Sublinear-Time Methods

by

Arsen Vasilyan

Submitted to the Department of Electrical Engineering and Computer Science
on March 20, 2024, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Our society increasingly relies on algorithms and data analysis to make critical decisions. Yet, almost all work in the theory of supervised learning has long relied on the following two assumptions:

1. D**istributional assumptions:** data satisfies conditions such as Gaussianity or uniformity.
2. **No distribution shift:** data distribution does not change between training and deployment.

While natural and often correct, these assumptions oftentimes do not hold. Yet, these assumptions are routinely made for giving theoretical guarantees for supervised learning algorithms. These guarantees can become null and void, should one of these algorithms be used in a setting where these assumptions do not hold. Overall, if critical decisions rely on theoretical reliability guarantees, incorrect assumptions can result in catastrophic failure.

The first part of this thesis shows how to mitigate this dependence. We introduce and develop testers which can alert a user if some assumptions are not satisfied. Leveraging insights from the area of property testing, the first part of this thesis constructs such testers for a number of well-studied function classes, addressing distributional assumptions and distribution shift.

The second part of this thesis shows how insights from sublinear-time algorithms can also be used to make learning algorithms more runtime-efficient. We show that sublinear-time local algorithms, capable of deriving partial solutions by examining only a fraction of the input, can be used as a powerful primitive to resolve problems in learning theory.

Thesis Supervisor: Jonathan Kelner
Title: Professor of Applied Mathematics

Thesis Supervisor: Ronitt Rubinfeld
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I am profoundly grateful to my PhD advisors, Ronitt Rubinfeld and Jonathan Kelner, for their unwavering support, guidance, and mentorship throughout my doctoral journey. They not only imparted invaluable knowledge but also instilled in me the skills necessary to navigate the complexities of research. From selecting research problems to presenting my work, their wisdom has been instrumental in shaping my academic path. I am especially grateful to Ronitt, with whom I have had the privilege of working since my undergraduate years.

I extend my sincere appreciation to Sam Hopkins for his invaluable contributions as a member of my thesis committee, offering invaluable insights and guidance.

I am indebted to Adam Klivans for his exceptional mentorship and collaboration, which significantly enriched the research presented in this thesis.

My heartfelt thanks go to my esteemed peers (in alphabetical order) Aravind Gollakota, Jane Lange, Konstantinos Stavropoulos for their significant contributions to the work discussed in this thesis. Furthermore, I learned a lot from all my other collaborators who worked with me on various research projects outside of this thesis. Listed in alphabetical order, their names are Sepehr Assadi, Gautam Chandrasekaran, Ilan Cohen, Talya Eden, Alon Eden, Khashayar Gatmiry, Surbhi Goel, Tim Hsieh, Esty Kelman, Pravesh Kothari, Frederic Koehler, Vasilis Kontonis, Ephraim Linder, Aleksander Madry Jeet Mohapatra, Ted Pyne, Sofya Raskhodnikova, Dhruv Rohatgi, Abhishek Shetty, Aaron Sidford, Jakub Tetek, Kevin Tian, Shih-Yu Wang, Ning Xie and Jeff Xu.

I extend my appreciation to my housemates Rahul Ilango and Shyam Narayanan, as well as my officemates Agnes Villanyi, Hannah Lawrence, and Sabrina Drammis, for their support and camaraderie. I am also grateful to the entire theory group at MIT for their stimulating environment and friendship. Special thanks to Anders Aamand, Shayan Akmal, Maryam Aliakbarpour, Ainesh Bakshi, Kiril Bangachev, Shankha Biswas, Matthew Brennan, Leo de Castro, Justin Chen, Sitan Chen, Lily Chung, Kristian Georgiev, Noah Golowich, Brice Huang, Vardis Kandiros, Surya Mathialagan, Klara Mundilova, Sandeep Silwal, Stefan Tiegel, Neekon Vafa, Nicole Wein, Yinzhan Xu, Rachel Zhang, and others for their contributions and support during various events and gatherings.

I owe a debt of gratitude to my friends outside of academia for their unwavering support, enriching conversations and eye-opening experiences. Special thanks to Janak Agarwal, Anuj Apte, John Friedman, Charles Fu, Chris Hillenbrand, Vardges Mambreyan, Tugsuu Manlaibaatar, Alexandra Martirosyan, Gevorg Martirosyan, Marta Maznin, Joey Muller, Jeet Mohapatra, Khachatur Nazaryan, Alex Patton, Debaditya Pramanik, James Rowan, Tomohiro Soejima, Vienna Thomas, Rona Wang, Henry Wu and others for their friendship and encouragement.

Last but not least, I am profoundly grateful to my family for their unconditional love, unwavering support, and encouragement throughout this journey. Without their support, this thesis would not have been possible. I extend my heartfelt thanks to my parents, Ashot Vasilyan and Irina Boyakhchyan, and to my grandparents, Efrem Boyakhchyan, Galina Boyakhchyan, Anahit Vasilyan and Moris Vasilyan for their invaluable role in shaping my life and academic pursuits.

Introduction

This thesis explores methodologies for improving learning algorithms, focusing on reliability and efficiency. The first part of this thesis concentrates on testing the assumptions that underpin learning algorithms. By drawing upon insights from sublinear-time distribution-testing literature, this section presents methods for testing these assumptions, enhancing the reliability of learning processes.

The second portion of the thesis delves into the realm of monotone function learning, aiming to address longstanding gaps in our understanding. Leveraging sublinear-time local algorithms, the second part of this thesis uncovers novel insights into this domain. We show that sublinear-time local algorithms, capable of deriving partial solutions by examining only a fraction of the input, can be used as a powerful primitive to resolve problems in learning theory.

Overall, this thesis shows that insights from sublinear-time algorithms can help improve learning algorithms by making them faster and more reliable.

Part 1: Ensuring Reliability of Learning Algorithms.

Our society increasingly relies on algorithms and data analysis to make critical decisions. Yet, almost all work in theory of supervised learning has long relied on the following types of assumptions:

1. **Distributional assumptions:** data satisfies conditions such as Gaussianity or uniformity over d-bit strings.

2. **No distribution shift:** when the classifier is deployed, data comes from the same distribution as the training examples.

While natural and often correct, these assumptions oftentimes do not hold. Yet, these assumptions are routinely made for giving theoretical guarantees for supervised learning algorithms. As a result, these guarantees can become null and void, should one of these algorithms be used in a setting where these assumptions are incorrect. Overall, if critical decisions rely on theoretical reliability guarantees, incorrect assumptions can result in catastrophic failure.

While, as it was known, completely eliminating the dependence on assumptions is provably impossible, the first part of this thesis shows how to mitigate this dependence. We introduce and develop testers which can alert the user if some assumptions are not satisfied. Conversely, if the

tester finds a specific dataset to be satisfactory, then a user can be confident in the guarantee given by the learning algorithm. Thus, having such a tester enables a user to make sure that a learning algorithm is safe to apply to a specific dataset.

Leveraging insights from the area of sublinear-time distribution testing, the first part of this thesis constructs such testers for a number of well-studied function classes, addressing distributional assumptions and distribution shift.

Testing distributional assumptions: In Chapters 1 and 2 we focus on the distributional assumptions made by **agnostic learning** algorithms. Agnostic learning is a well-studied setting in learning theory, in which the learning algorithm should find a classifier that performs as well as the best classifier in a given hypothesis class $\mathcal{F}$. Notably, an agnostic learning algorithm is robust to an adversary corrupting a subset of labels in the data. Almost all algorithms in this area rely on distributional assumptions on the high-dimensional data these algorithms receive, such as Gaussianity or uniformity over d-bit binary strings [KKMS08, OS06, BOW08, KOS08, GS10, Kan10, Wim10, DHK$^+$10, CKKL12, ABL14, FK15, BCO$^+$15, DKK$^+$21]. (Note that such distributional assumptions are unavoidable due to known hardness results such as [GR06, FGKP06, Dan16].)

Reliance on distributional assumptions comes with risks. Reliability guarantees become null and void, should one of these algorithms be accidentally misapplied to a setting where these assumptions do not hold. For example, this could lead to an adversary making the learning algorithm fail spectacularly by corrupting only a tiny fraction of data labels.

Motivated by this issue, in Chapter 1, which is based on [RV23], we introduce the **tester-learner** framework for agnostic learning. In our new framework, we require that an agnostic learning algorithm comes together with a tester, which can alert a user if some assumptions are not satisfied. Conversely, if the tester finds a specific dataset to be satisfactory, then a user can be **confident in the guarantee** given by the learning algorithm. Thus, having such a tester enables a user to make sure that the agnostic learning algorithm is safe to apply to a specific dataset.

Although our work draws heavily on the ideas from traditional sublinear-time distribution testing, off-the-shelf distribution testing algorithms are too slow for our purposes. The reason is that previous testers for properties of distributions such as uniformity or Gaussianity have run-times that are exponential in the dimension, which is dramatically slower than many agnostic learning algorithms. These exponential run-times are provably necessary for testers that are defined with respect to standard distance measures such as total variation distance or earth-mover distance. See text [Can22] for a survey of such testers. Our tester-learner framework sidesteps these lower bounds as it does not require us to check that the distribution of the data is close to uniform or Gaussian with respect to these standard distance measures. Instead, our framework only requires us to make sure the distribution is sufficiently similar to the Gaussian or the uniform distribution for agnostic learning algorithms to be successful.

In Chapter 1 we show that for the class of **linear classifiers**, one can indeed obtain a tester-learner pair with combined run-time of the same order as best distribution-specific agnostic learning algorithms. Note that Follow-up work by Gollakota, Klivans and Kothari improved our run-time and showed that fast tester-learner pairs exist for a wider variety of function classes, including

AC0 circuits and intersections of linear classifiers [GKK23].

In Chapter 2, which is based on [GKSV24] and [GKSV23], we further study tester-learner pairs for linear classifiers that run in **polynomial time in all parameters**[1] (also see the work [DKK+23]). In the distribution-specific setting, such an algorithm was first given by the celebrated work of [ABL14]. We complement [ABL14] by giving a tester-learner pair for this problem. To achieve this, we expand our testing toolkit by going beyond the moment-matching approach used in Chapter 1 and [GKK23], which checks that low-degree polynomials cannot distinguish the data distribution from the assumed distribution. Instead, in Chapter 2 we introduce direction-aware testers that partition the d-dimensional space into a series of regions along a vector obtained by running the algorithm of [DKTZ20b], which is based on non-convex stochastic gradient descent. Then we perform the moment-matching test in each of these regions separately (instead of in the whole of d-dimensional space at once). This approach yields a tester-learner pair with combined run-time polynomial in all parameters.

Another novel property possessed by the tester-learner pair in Chapter 2 is its ability to handle a whole family of assumptions: the family of strongly log-concave distributions (in contrast, Chapter 1 only handles specific distributions such as the standard Gaussian). Furthermore, the tester-learner pair we give in Chapter 2 satisfies what we call the **universality** property: the testing algorithm is guaranteed to accept every single distribution in the family of allowed distributions (without knowing in advance which particular one of these distributions it will face). A crucial new ingredient for our testing algorithm is the use of a sum-of-squares relaxation.

Addressing distribution shift. In Chapter 3, which is based on [KSV23] we study how to mitigate the effects of **distribution shift**. In learning theory, it is commonly assumed that the training dataset comes from the same distribution as the data we see during deployment. Yet, due to a distribution shift, this assumption might not hold, which can lead to incorrect predictions. Dealing with distribution shift is a big challenge in machine learning. For example, classifiers trained on data from one hospital often fail to generalize to other hospitals [ZBL+18, WOD+21, TCK+22]. Inspired by our previous work in testing assumptions, in Chapter 3 we develop a new framework called **Testable Learning with Distribution Shift** (TDS learning) for addressing this issue. The goal of our work is developing learning algorithms that allow their user to make sure that the performance of a learning algorithm is not deteriorating due to a distribution shift.

In Chapter 3, we consider the following setting. We are given a training dataset S_1, which for example might contain data from the Gaussian distribution which are labelled by an unknown function f in the function class. We are also given a new dataset S_2 of unlabelled data points which we need to classify. The concern is that even if a learning algorithm produces a classifier that performs very well on the distribution from which S_1 is drawn, it might perform terribly on S_2 due to a distribution shift.

[1] To achieve such run-times, we target the *semi-agnostic* guarantee, as is also done in [ABL14]. This guarantee is slightly less strict than the agnostic learning guarantee, yet it still provides robustness to adversarial label corruptions. (Fully agnostic learning of linear classifiers is believed to not be achievable in time polynomial in all parameters. This is due to statistical query lower bounds, as well as reductions from cryptographic assumptions.)

Whenever distribution shift makes it unsafe to use the classifier given by the learning algorithm on the unlabeled dataset S_2, the TDS learning framework requires the learning algorithm to alert the user. In contrast, if no distribution shift has taken place, then the TDS learning framework requires the learning algorithm not to raise such an alert. In this case, the user can be confident that the classifier given by the learning algorithm is indeed **safe to deploy** on the dataset S_2.

Although domain-adaptation has been studied previously, most previous work such as [BDBCP06, BDBC$^+$10, MMR09] bounds the error on S_2 in terms of some notion of distance between the distributions giving rise to S_1 and S_2. These distances seem computationally difficult to evaluate, as they involve enumeration of all elements of the function class. (Additionally, off-the-shelf distribution testers have run-times much higher than most learning algorithms. See text [Can22] for a survey of such testers.)

In Chapter 3 we give TDS learning algorithms that are not only sample-efficient but also **computationally efficient**. We handle a number of high-dimensional function classes, including linear classifiers, intersections of linear classifiers, decision trees and low-depth formulas. Our TDS learning algorithms work when training dataset S_1 comes from the Gaussian distribution or the uniform distribution on n-bit strings. To accomplish this, we develop a novel method based on showing that functions in a given function class can be sandwiched between pairs of low degree polynomials that are close in L_2-distance. We prove the existence of such pairs of L_2-sandwiching polynomials for a wide range of function classes, by building on techniques from **pseudorandomess** used in [DGJ$^+$10] and [GOWZ10]. We then show that the existence of these L_2-sandwiching polynomials can be used to achieve efficient TDS-learning for the aforementioned function classes by introducing our Transfer Lemma, which allows us to ensure a good performance on the set S_2 by checking that the low-degree moments of the set S_2 approximately match those of S_1.

Furthermore, for TDS learning of linear classifiers under the Gaussian distribution, we give an algorithm that does provably better than any algorithm based on L_2-sandwiching polynomials. To accomplish this, we combine the moment-matching approach with a method inspired by literature on active learning. Additionally, we extend the notion of TDS learning into the agnostic learning setting. By leveraging insights from our work on tester-learner pairs [GKSV23], we give agnostic TDS learning algorithms for homogeneous linear classifiers that runs in polynomial time in all parameters.

Part II: Closing Computational-to-Statistical Gaps via Sublinear-Time Correction.

For numerous problems in learning theory and high-dimensional statistics, there are computational-to-statistical gaps in our understanding. In other words, for such problems there are statistically efficient algorithms with low data consumption, and yet no computationally efficient algorithms are known. In Chapters 4 and 5 we give explore a new algorithmic method for addressing these gaps, which is based on sublinear-time Local Computation Algorithms.

Prior to our work in Chapters 4 and 5, a longstanding computational-to-statistical gap existed for many problems related to **high-dimensional monotone functions**. A binary-valued function

on d-bit strings is called monotone if increasing the value of one of the input bits from 0 to 1 can only cause the value of the function to increase. In Chapters 4 and 5, we consider the following fundamental problems about monotone functions:

1. **Proper**[2] **learning:** Given uniform samples labelled by a monotone function f, find a monotone function that approximates[3] f up to error ϵ.

2. **Proper agnostic learning:** Given uniform samples labelled by an arbitrary function f, find a monotone function that approximates f best among all monotone functions (up to error ϵ).

3. **Distance approximation:** Given uniform samples labelled by an arbitrary function f, approximate the distance of f to the monotone function closest to f (up to error ϵ).

Since [BT96], it was known that all the problems above can be solved using only $2^{\tilde{O}(\sqrt{d}/\epsilon)}$ samples, yet prior to our work no known algorithms for any of these tasks ran in time $2^{o(d)}$. This was true even for algorithms that could additionally query the function f on arbitrary inputs of their choice.

In Chapters 4 and 5, we close this computational-to-statistical gap by giving algorithms for each of the problems above that run in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. One of our techniques is a novel use of sublinear-time **Local Computation Algorithms** (LCAs) for the maximal matching problem on graphs.

In Chapter 4, which is based on [LRV22], we leverage these techniques in order to obtain an efficient **monotonicity corrector** that transforms a given binary-valued function into a monotone function by changing as few of its values as possible. An important property of our corrector is that it is **local**, i.e. the value of the corrected function on a specific input x is deduced by only evaluating the original function on a relatively small number of elements in the neighborhood of x. Applying our local corrector to the non-monotone hypothesis produced by the algorithm of [BT96], we give a proper learning algorithm for monotone functions that runs in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. Although local monotonicity correctors were studied previously [ACSL08, ACSL07, SS10a, CGR13, AJMR14], all existing correctors ran in time $2^{\Omega(d)}$, which is too large to yield improved algorithms for proper learning of monotone functions.

Chapter 5, which is based on [LV23], improves on Chapter 4 and obtains a run-time of $2^{\tilde{O}(\sqrt{d}/\epsilon)}$ also for the task of proper **agnostic** learning of monotone functions, as well as the task of approximating the distance of a function to monotone. The algorithm for proper agnostic learning of monotone functions proceeds in two stages.

The first stage runs a low-degree polynomial L_1-regression, while also satisfying a number of constraints that make sure that a polynomial is close to a monotone function in L_1 norm. Although the number of these additional constraints we impose is exponentially large, in Chapter 5 we make sure they are all satisfied simultaneously. We do this by giving a separation oracle that, given a polynomial violating one of these constraints, provides a linear certificate that one of

[2]The word "proper" is used to distinguish this setting from the so-called **improper** learning setting, in which the learning algorithm is allowed to output any hypothesis (not necessarily one in the hypothesis class).

[3]In this section, we define the distance between two functions as the fraction of the elements in their domain on which the two functions disagree.

these constraints is violated. This allows us to leverage the Ellipsoid algorithm in order to keep all these constraints satisfied simultaneously, because the Ellipsoid algorithm can solve systems of exponentially many constraints, as long as a separation oracle for these constraints is provided.

In Chapter 5, we implement the separation oracle by building on the LCA-based techniques of Chapter 4. To do this, we first show that a high-weight matching M of monotonicity-violating pairs of elements can certify that a given polynomial P is far from any real-valued monotone function. We then show that such matching M certifies that not only P is far from any real-valued monotone function, but also that each polynomial beyond a certain hyperplane in the space of polynomials is far from any real-valued monotone function. Finding such a hyperplane yields exactly a linear certificate that our separation oracle needs to output, and we show how to find it in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. Note that this run-time is dramatically faster than even writing our the whole matching M, as it typically has a size as large as $2^{\Omega(d)}$. We accomplish this by first employing LCAs to obtain local access to such a matching M. Having local access to this matching M, we are able to accurately "learn" the hyperplane in the space of polynomials beyond which it certifies large distance to monotonicity via accessing the matching M on $2^{\tilde{O}(\sqrt{d}/\epsilon)}$ random d-bit strings. We show that this information suffices to estimate this hyperplane accurately.

Once the Ellipsoid algorithm terminates, our algorithm in Chapter 5 obtains a polynomial P that both fits the target function well, and is also close to monotone in the L_1 norm. The second stage of our algorithm transforms this polynomial P into a monotone function that is close to P. In order to achieve this, we extend the monotonicity corrector of Chapter 4 to work with general real-valued functions (the original monotonicity corrector of Chapter 4 only works with binary-valued functions).

Contents

List of Figures

Part I

Ensuring Reliability of Learning Algorithms.

Chapter 1

Testing distributional assumptions of learning algorithms.

1.1 Chapter Overview.

1.1.1 Motivation.

Suppose one wants to learn from independently distributed example-label pairs, but some unknown fraction of labels are corrupted by an adversary. The well-studied field of **agnostic learning** seeks to develop learning algorithms that are robust to such corruptions. (See [BLMT22] for more on how exactly agnostic learning algorithms yield algorithms that are resilient to adversarial noise in labels.) Agnostic learning can be notoriously harder than standard learning (see for example [HJLT96, GR06, FGKP06, Dan16]). Nevertheless, there are many important **high-dimensional** function classes that do have fast agnostic learning algorithms, including halfspaces, convex sets and monotone Boolean functions. However, these learning algorithms make strong assumptions about the underlying distribution on examples , such as Gaussianity or uniformity over $\{0,1\}^d$.

Thus, to be confident in such a learning algorithm one needs to be confident in the distributional assumption. In some cases, users can attain confidence in their distributional assumptions by creating their own set of examples which conform to the distribution, and querying labels for these examples. Yet, this approach requires query access, which is often unavailable. Is there a way to ascertain that the examples are indeed coming from a distribution for which the learning algorithm will give a robust answer?

We propose to systematically study the design of **tester-learner pairs** $(\mathcal{A}, \mathcal{T})$, such that **tester** $\mathcal{T}$ **tests the distributional assumptions of agnostic learner** $\mathcal{A}$. In other words, the tester-learner pair is to be designed such that if the distribution on examples in the data pass the tester, then one can *safely use the learner on the data*. By considering the most basic requirements that such a pair ought to satisfy, we propose a new framework that makes the following end-to-end requirements on a tester-learner pair $(\mathcal{A}, \mathcal{T})$:

- **Soundness:** For any example-label distribution, it should be unlikely that simultaneously

(i) the tester $\mathcal{T}$ accepts but (ii) the learner $\mathcal{A}$ outputs something not satisfying the agnostic learning guarantee.

- **Completeness:** If the distribution on examples conforms to the distributional assumption, tester $\mathcal{T}$ will likely accept.

- The performance of the tester-learner pair is judged by the combined run-time of $\mathcal{A}$ and $\mathcal{T}$.

See Section 1.2.2 for the fully formal definition and see Subsection 1.1.3 for more comments.

We emphasize that assumptions on the distribution of examples are in fact made in a very large number of works on agnostic learning. Here is an incomplete list of such papers that only scratches the surface: [KKMS08, OS06, BOW08, KOS08, GS10, Kan10, Wim10, HKM10, DHK$^+$10, CKKL12, ABL14, DSFT$^+$14, FV15, FK15, BCO$^+$15, CGG$^+$17, FKV17, DKK$^+$21]. The reason for this ubiquity of distributional assumptions in high-dimensional agnostic learning is that with no assumption at all on the distribution the task of agnostic learning is usually intractable[1]. Hence, it is important to understand to what extent these distributional assumptions can be tested.

Perhaps surprisingly, in spite of how natural this definition is, nothing was previously known on how efficiently it can be achieved for various well-studied problems. The gamut of open possibilities included the most optimistic one: that for all these problems one can test the assumption with very small overhead relative to the existing agnostic learning algorithms. It also included the most pessimistic one: that for all these problems one can test the assumption only at a very steep additional cost in terms of run-time. We note that such steep additional cost would indeed be payed if one were to use existing identity testers of d-dimensional distributions, as these testers have run-times of $2^{\Omega(d)}$ (see below for more information on this).

We commence the charting of the landscape of these possibilities. This run-time qualitatively matches the run-time of $d^{\tilde{O}(1/\epsilon^2)}$ [KKMS08, DGJ$^+$10] achieved by the best algorithm[2] and the statistical query lower bound of $d^{\Omega(1/\epsilon^2)}$ by [GGK20, DKZ20, DKPZ21]. We also go beyond spherically-symmetric distributions, and give a tester-learner pair for halfspaces under the uniform distribution on $\{0,1\}^d$ with combined run-time of $d^{\tilde{O}(1/\epsilon^4)}$. Here also, the run-time qualitatively matches the run-time of $d^{\tilde{O}(1/\epsilon^2)}$ [KKMS08, DGJ$^+$10] achieved by the best algorithm. Additionally, we remark that positive results in our framework extend to function classes beyond halfspaces and, as a proof of concept, we give a simple tester-learner pair for agnostically learning decision lists[3]

[1]For example (i) The task of learning indicators of convex sets over $\mathbb{R}^d$ cannot be achieved with finite number of samples if nothing is assumed about the distribution. If the distribution is assumed to be Gaussian, this task can be achieved with run-time of $d^{\tilde{O}(\sqrt{d}/\epsilon^4)}$ [KOS08]. (ii) If one is unwilling to make any distributional assumption, no agnostic learning algorithm for halfspaces with run-time of $2^{o(d)}$ is known despite decades of research (also see [GR06, FGKP06, Dan16] for some known hardness results). However, as we mentioned if the examples are distributed according to the standard Gaussian, a dramatically faster run-time of $d^{\tilde{O}(1/\epsilon^2)}$ is achievable [KKMS08, DGJ$^+$10].

[2]In this chapter we focus on agnostic learning algorithms achieving error opt $+\,\epsilon$. See Chapter 2 for the study of tester-learner pairs for the weaker guarantee of $O(\text{opt})+\epsilon$, which is sometimes referred to as "semi-agnostic" learning.

[3]For this example, a *decision list* is a special case of a decision tree corresponding to a path. More formally, for some ordering of the variables $x_{\pi(1)},\ldots,x_{\pi(d)}$, values $v_1,\ldots,v_n$ and bits $b_1,\ldots,b_n$, a decision list does the following: For $i=1$ to d, if $x_{\pi(i)}=b_{\pi(i)}$ output $v_{\pi(i)}$, else continue. A more general definition is given in [Riv87].

under uniform distribution on $\{0,1\}^d$ (see Section 1.9.6). Also see [HJLT96] for some intractability results on distribution-free learning of decision lists.

On the other hand, for some other natural problems, we show that the most pessimistic scenario holds and the additional requirement of testing the distributional assumption comes at a steep price in terms of run-time. Specifically:

- A well-known algorithm of [KOS08] agnostically learns convex sets under the Gaussian distribution with a run-time of $d^{\tilde{O}(\sqrt{d}/\epsilon^4)}$. We show that if a tester $\mathcal{T}$ tests the distributional assumption of this algorithm, then $\mathcal{T}$ has run-time of $2^{\Omega(d)}$. More generally, **any** tester-learner pair for this task requires $2^{\Omega(d)}$ run-time combined.
- A well-known algorithm of [BT96, KKMS08] agnostically learns monotone Boolean functions under uniform distribution over $\{0,1\}^d$ with a run-time of $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon^2}\right)}$. We show that if a tester $\mathcal{T}$ tests the distributional assumption of this algorithm, then $\mathcal{T}$ has run-time of $2^{\Omega(d)}$. Again, **any** tester-learner pair for this task requires $2^{\Omega(d)}$ run-time combined.

We emphasize that these lower bounds exhibit natural problems where there is a dramatic gap between standard agnostic learning run-time and the run-time of the best tester-learner pair. Therefore, there is provably no general method that allows one to automatically convert standard agnostic learning algorithms into tester-learner pairs with low run-time overhead.

Additionally, lower bounds for tester-learner pairs can imply lower bounds for standard agnostic learning: Specifically, our lower bounds imply that agnostic learning of monotone functions under distributions $\frac{1}{2^{d^{0.99}}}$-close[4] to $d^{0.99}$-wise independent distributions requires $2^{\Omega(d)}$ run-time. The reason is that by [OZ18, AAK+07, AGM03] one can test $d^{0.99}$-wise independence up to error $\frac{1}{2^{d^{0.99}}}$ in time $2^{\tilde{O}(d^{0.99})}$, and therefore the existence of such an algorithm would contradict our general lower bound for tester-learner pairs. As there are $2^{\tilde{O}(\sqrt{d}/\epsilon^2)}$ time learners for monotone functions over the uniform distribution [BT96, KKMS08], this lower bound highlights the sensitivity of agnostic learners to the assumption on the input distribution.

Distribution testing perspective. Existing work on identity testing of d-dimensional distributions has focused on testing with respect to very strict distance measures (such as TV distance, earth-mover distance and other distance measures coming from probability theory). On one hand this yields strong general-purpose guarantees on distributions accepted by the tester – it is hard to think of a situation where closeness in TV distance is unsatisfactory. On the other hand, in d dimensions this leads to run-times of $2^{\Omega(d)}$. As a concrete example, distinguishing the uniform distribution over $\{0,1\}^d$ from a distribution that is ϵ-far from it in total variation distance requires a run-time of $\Theta\left(\frac{1}{\epsilon^2}2^{d/2}\right)$ (see text [Can22]).

Yet, run-times of $2^{\Omega(d)}$ can be prohibitive. Indeed, as we explained above, the theory of d-dimensional agnostic learning aims at developing algorithms with run-times of $2^{o(d)}$ or even $d^{O_\epsilon(1)}$.

[4] In total variation distance.

If one were to combine these algorithms with a $2^{\Omega(d)}$-run-time distribution tester, the total run-time would rise precipitously.

From the distribution testing perspective, this chapter studies **application-targeted testers** that, in favor of much faster run-time, forgo the general-purpose guarantees provided by these strict distance measures. The application domain which this chapter considers is the testing of distributional assumptions made by agnostic learning algorithms. Here, the application-targeted testers are developed with a view towards special-purpose guarantees sufficient to ensure that the learning algorithms are still robust. For some problems in this domain – this chapter shows – the use of general-purpose testers can indeed be circumvented, with a dramatic gain in run-time.

In general, surprisingly little is known about such application-targeted testers and we hope more application-targeted distribution testers can be developed for other domains.

Brief comparison with distribution-free agnostic learning. Recall that an agnostic learning algorithm is **distribution-free** if it succeeds regardless of the distribution on examples. Designing such algorithms has proven to be intractable for many function classes (see for example [HJLT96, GR06, FGKP06, Dan16]). This intractability has prompted the study of agnostic learning algorithms under distributional assumptions.

The framework we introduce in this chapter is intermediate between distribution-free agnostic learning and agnostic learning under a distributional assumption. While the learning algorithm is not required to satisfy the agnostic learning guarantee under every single distribution on example, the testing algorithm needs to alerts us whenever the learning algorithm does fail to satisfy this guarantee.

Incidentally, when using a tester-learner pair, whenever the testing algorithm rejects, the user can choose to then run a slow distribution-free agnostic learning algorithm. Overall, this strategy yields a learning algorithm that always satisfies the agnostic guarantee, and additionally runs fast whenever the distributional assumption does hold, thereby adapting to the distribution on examples.

The followup work of [GKK23]. A follow up work [GKK23] builds on an earlier version of this chapter, which had been made available to them. [GKK23] develops novel techniques for the design and analysis of tester-learner pairs that leverage connections with the notion of fooling a function class from the field of pseudorandomness. This allows [GKK23] to

- Give tester-learner pairs for more general function classes, such as intersections of halfspaces.
- Handle more general classes of distributional assumptions, such as strictly subexponential distributions in $\mathbb{R}^d$ and uniform over $\{0,1\}^d$.
- Present a new connection between the notion of tester-learner pairs and Rademacher complexity.
- Improve on our run-time for tester-learner pairs for halfspaces under the Gaussian distribution on $\mathbb{R}^d$. Specifically, they give a bound of $d^{\tilde{O}(1/\epsilon^2)}$ which improves upon our bound

of $d^{\tilde{O}(1/\epsilon^4)}$. Their tighter bound also matches the known statistical query lower bounds [GGK20, DKZ20, DKPZ21].

We would like to note that Theorem 4 (tester-learner pairs for halfspaces under the uniform distribution on $\{0,1\}^d$) is concurrent work with [GKK23] (they give a faster run-time of $d^{\tilde{O}(1/\epsilon^2)}$ for this problem and also give more general results as explained above). The earlier version of this chapter (which they build upon) already contained the other results presented in our current version, i.e. (i) the definition of tester-learner pairs (ii) the tester learner pair for half-spaces under the Gaussian distribution with run-time $d^{\tilde{O}(1/\epsilon^4)}$ (Theorem 2) (iii) the intractability results for tester-learner pairs in Theorems 5 and 6.

Further direct follow-up is presented in Chapter 2.

1.1.2 Our techniques.

Function class	Halfspaces	Halfspaces
Distributional assumption	Standard Gaussian in $\mathbb{R}^d$	Uniform on $\{0,1\}^d$
Standard agnostic learning *run-time* from literature	$d^{\tilde{O}(1/\epsilon^2)}$ [KKMS08, DGJ+10]	$d^{\tilde{O}(1/\epsilon^2)}$ [KKMS08, DGJ+10]
Standard agnostic learning *intractability* from literature	$d^{\Omega(1/\epsilon^2)}$ statistical queries [GGK20, DKZ20, DKPZ21]	We are not aware of published intractability results in this setting.
Examples needed for testing assumption in TV distance	infinite	$\Theta\left(\frac{1}{\epsilon^2}2^{d/2}\right)$ (see text [Can22])
The run-time of our tester-learner pair	$d^{\tilde{O}\left(1/\epsilon^4\right)}$	$d^{\tilde{O}\left(1/\epsilon^4\right)}$

Table 1.1: Summary of our algorithms and relevant previous work.

We summarize the contributions in this chapter and relevant background in Table 1.1 on page 18 and Table 1.2 on page 19.

Tester-learner pair for agnostically learning halfspaces under Gaussian distribution We first give an overview of our tester-learner pair $(\mathcal{A}, \mathcal{T})$ with combined run-time of $d^{\tilde{O}(1/\epsilon^4)}$ for the class of half-spaces with respect to standard Gaussian distribution. We also discuss the techniques we use to analyze it. See Sections 1.3, 1.5 and 1.6 for complete details.

A natural first approach would be to try to take advantage of the literature on testing and learning distributions. However, almost all results we are aware of on testing and learning high-

Function class	Convex sets	Monotone functions
Distributional assumption	Standard Gaussian in $\mathbb{R}^d$	Uniform on $\{0,1\}^d$
Standard agnostic learning *run-time* from literature	$d^{\tilde{O}(\sqrt{d}/\epsilon^4)}$ [KOS08]	$2^{\tilde{O}(\sqrt{d}/\epsilon^2)}$ [BT96, KKMS08]
Standard agnostic learning *intractability* from literature	$d^{\Omega(\sqrt{d})}$ [KOS08]	$2^{\tilde{\Omega}(\sqrt{d})}$ [BCO+15]
Examples needed for testing assumption in TV distance	infinite	$\Theta\left(\frac{1}{\epsilon^2}2^{d/2}\right)$ (see text [Can22])
Our lower bound for combined run-time of a tester-learner pair	$2^{\Omega(d)}$	$2^{\Omega(d)}$

Table 1.2: Summary of our intractability results and relevant previous work.

dimensional distributions (without assuming the distribution already belongs to some highly restricted family as in [CM13]) require a number of samples that is exponentially large in the dimension. It follows from well-known techniques that Gaussianity over an infinite domain cannot be tested with respect to total variation distance in finite samples. Potentially, one could obtain a tester-learner pair for Gaussianity with respect to the earth-mover distance via the tester[5] of [BNNR11], yielding a tester of run-time $2^{\tilde{O}(d)}$. However one can see that, in earth-mover distance, no significantly better (i.e. $2^{o(d)}$) bound can be obtained[6]. Such enormous run-times far exceed the run-times that can be achieved for agnostically learning halfspaces.

Previously it was known that half-spaces are well-approximated with low-degree polynomials relative to the Gaussian distribution. A key step in our analysis is showing that this is the case even relative to distributions whose low-degree moments approximately match those of a Gaussian. One of our ideas is to start with a proof of the exact Gaussian case and modify it so it only relies on low-degree properties of the distribution. We are aware of three distinct proofs of this exact Gaussian case in the literature:

1. The method of [KKMS08] that uses specific facts about Hermite polynomials.

2. The noise sensitivity method of [KOS08]. This method also uses Hermite polynomials to argue that functions that tend to be stable to perturbations of their input tend to be well-approximated by low-degree polynomials.

[5]This tester requires that the distribution is confined to a box $[-B,B]^d$, but this by itself is not a devastating problem, since most of probability mass of a Gaussian is confined to such a box.

[6]Even when truncating the distribution to a box around the origin.

3. The method of [DGJ$^+$10] that, in order to approximate a halfspace $\mathrm{sign}(\boldsymbol{v}\cdot\boldsymbol{x}+\theta)$, constructs a polynomial $P(\boldsymbol{v}\cdot\boldsymbol{x})$ that approximates this halfspace tightly for values of $|\boldsymbol{v}\cdot\boldsymbol{x}|$ that are not too large. It is then argued that large values of $|\boldsymbol{v}\cdot\boldsymbol{x}|$ do not contribute much to the total L_1 error of the polynomial because its contribution is weighted by a rapidly decaying Gaussian weight.

As Hermite polynomials are the unique family of polynomials orthogonal under the Gaussian distribution, the proof strategies of [KKMS08] and [KOS08] seem highly specialized to the distribution being exactly Gaussian. Because of this, a method similar to the one of [DGJ$^+$10] is the one serving as our starting point.

This method needs to be modified in a thoroughgoing way in order to rely merely on the low-degree moments of the distribution being close to those of Gaussian. For instance, a very easy-to-show property of the d-dimensional standard Gaussian distribution is its anti-concentration when projected on any direction. This property becomes much less obvious once one is only promised that low-degree moments of the distribution are close to those of Gaussian, which is something we do show. We note that this step of our proof is similar in spirit to the work of [KKK19b] that introduces a notion of low-degree certified anti-concentration and shows it for various distributions. Our proofs use extensively tools from polynomial approximation theory.

Given these ideas, our tester-learner pair does the following. The tester estimates the low-degree moments of the distribution and compares them to the corresponding moments of the standard Gaussian. It follows then that halfspaces are well-approximated by low-degree polynomials with respect to this distribution. The learning algorithm takes advantage of this by performing low-degree polynomial L_1 regression similar to the one used in [KKMS08].

A technical complication, which we deal with, is that both our tester and learner work with a truncated version of the distribution. In other words, they discard the examples whose coordinates are too large. This guarantees to us that we can actually produce estimates for the moments of the truncated distribution (if distribution is not truncated, moments could even be infinite).

Note that our arguments use strongly the fact that we are working with halfspaces and not with some arbitrary function class that is well-approximated by low-degree polynomials under the Gaussian distribution. This is due to how we use the concentration and anti-concentration properties of the distribution. In a certain sense this is necessary, as shown by our intractability results for indicators of convex sets. Even though these functions are also well-approximated by low-degree polynomials [KOS08], for them a similar method based on estimating low-degree moments will provably not succeed. This underscores that designing tester-learner pairs can be subtle and does not generally follow by mere extension of already existing analyses of agnostic learning algorithms.

Tester-learner pair for agnostically learning halfspaces under uniform distribution on $\{\pm 1\}^d$. We now discuss the techniques used to give our tester-learner pair for halfspaces under the uniform distribution on $\{\pm 1\}^d$. As we mentioned, the run-time we show here is $d^{\tilde{O}\left(1/\epsilon^4\right)}$ and this is concurrent work with [GKK23], who use other techniques. See Section 1.7 for complete details.

Our tester tests $\mathrm{poly}(1/\epsilon)$-wise independence of the input distribution with respect to the TV distance using [OZ18, AAK$^+$07, AGM03]. The learning algorithm uses the low-degree polynomial L_1 regression of [KKMS08]. To show that these two algorithms indeed form a valid tester-learner pair we show that every halfspace is well-approximated by a low-degree polynomial relative to any $\mathrm{poly}(1/\epsilon)$-wise independent distribution.

Suppose for a halfspace $\mathrm{sign}(\boldsymbol{v} \cdot \boldsymbol{x} + \theta)$ it is the case that the norm of the vector $\boldsymbol{v}$ is well-distributed among all the coordinates. Then, by Berry-Esseen theorem, for $\boldsymbol{x}$ that is uniform over $\{\pm 1\}^d$ the inner product $\boldsymbol{v} \cdot \boldsymbol{x}$ is distributed similarly to a Gaussian. Roughly, we use this to argue that if $\boldsymbol{x}$ is merely $\mathrm{poly}(1/\epsilon)$-wise independent then $\boldsymbol{v} \cdot \boldsymbol{x}$ has low-degree moments close to those of a Gaussian. This allows us to use methods similar to the ones we use to give tester-learner pairs for halfspaces under the standard Gaussian distribution.

Finally, we handle halfspaces $\mathrm{sign}(\boldsymbol{v} \cdot \boldsymbol{x} + \theta)$ for whom the norm of the vector $\boldsymbol{v}$ is not well-spread across all the coordinates. We use the *critical index* machinery of [DGJ$^+$10] to handle such halfspaces.

Intractability results. Finally, we discuss the techniques used to show that $2^{\Omega(d)}$ samples are required by (i) any tester-learner pair for learning indicator functions of convex sets under the standard Gaussian on $\mathbb{R}^d$ (ii) any tester-learner pair for learning monotone functions under the uniform distribution on $\{0,1\}^d$. See Section 1.8 for complete details.

From technical standpoint, we find these lower bounds surprising: The mentioned standard agnostic learning algorithms in these settings rely on low-degree polynomial regression. This suggests that testing low-degree moments of the distribution (as we did for halfspaces) ought to lead to the development of a fast tester-learner pair. Yet, the lower bounds show that this can not be done.

We now roughly explain how we prove these lower bounds. Let us focus on the lower bound for tester-learner pairs for convex sets under standard Gaussian distribution (the lower bound for monotone functions is similar). Take samples $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M$ from the standard Gaussian, and let D be the uniform distribution on $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$. The first idea is to show that the tester will have a hard time distinguishing D from the standard Gaussian if it uses much fewer than M samples. (Out actual argument also takes into account that the tester sees labels and not only examples.) The second idea is to show that (very likely over the choice of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M$) one can obtain, by excluding only a small fraction of elements from $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$, a subset Q of them such that no point in Q is in the convex hull of the other points in Q. Once we have such a set, we essentially[7] define our hard-to-learn convex set to be the convex hull of a random subset of Q, and this convex set will not contain any other elements of Q because no member of Q is in the convex hull of the rest. In this way, unless a learner has seen a large fraction of the elements in Q already, it has no way of predicting whether a previously unseen element in Q belongs to the random convex set. We note that our argument is somewhat similar to well-known arguments proving impossibility of

[7]This is an oversimplification, as one still needs to figure out what to do with elements outside Q. We show that, for all these elements, we can either include them into or exclude them from the convex set in such a way as to reveal no information about which of the points in Q were included in the convex set.

approximation of the volume of a convex set via a deterministic algorithm [BF86, Ele86].

1.1.3 Comments on the framework.

What about cross-validation? In case of realizable learning (i.e. you are promised there is no noise) a common approach to verifying success is via checking prediction error rate on fresh data and making sure it is not too high. Does this idea allow one to construct a tester $\mathcal{T}$ for the distributional assumption of some agnostic learner $\mathcal{A}$? Such tester would (i) run $\mathcal{A}$ to obtain a predictor $\widehat{f}$ (ii) test the success rate of $\widehat{f}$ on fresh example-label pairs (iii) accept or reject based on the success rate.

As was mentioned in the discussion of our intractability results, there cannot be a general low-overhead method of transforming standard agnostic learning algorithms into tester-learner pairs, because of our intractability results. Therefore, in particular, there cannot be such a method based on cross-validation.

Intuitively, the reason is the following. Suppose you run the learning algorithm, setting the closeness parameter ϵ to 0.01, then check the success of the predictor on fresh data and find that the generalization error is close to 0.25. This could potentially be consistent with the two following situations: (1) there is a function in the concept class with close to zero generalization error, but the learning algorithm gave a poor predictor due to a violation of the distributional assumption (2) the distributional assumption holds, but every function in the concept class has generalization error of at least 0.24. The *Soundness* criterion tells you that in case (1) you should reject, but the *completeness* criterion tells you that in case (2) you should accept. Overall, there is no way to tell from generalization error alone which of the two situations you are in, so there is no way to know if you should accept or reject.

Label-aware vs label-oblivious testers. We say the tester $\mathcal{T}$ is *label-aware* if it makes use of the labels given to it (and not only the examples). Otherwise, we call it *label-oblivious*. We feel that label-obliviousness makes a testing algorithm fit better with the existing literature on testing properties of distributions, because algorithms in this line of work decide to accept or reject a distribution based only on samples from it (and no side information such as labels). However, this condition is not strictly necessary for verifying success. Due to these considerations, our impossibility results are against more general label-aware testers, while the tester given in this chapter is label-oblivious.

1.1.4 Related work.

Agnostic learning under distributional assumptions using low-degree polynomial regression. Since the introduction of the agnostic learning framework [Hau92, KSS94a] there has been an explosion of work in agnostic learning. Making assumptions on the distribution on examples has been ubiquitous in this line of work. So has been the use of low-degree polynomial regression as one of the main tools. Previous to the work of [KKMS08], there existed an extensive body

of work on using low-degree polynomial regression for learning under distributional assumptions, including [LMN93, AM91, FJS91, Man92, BT96, KOS02]. The work of [KKMS08] building on [KSS94a] proposed to use low-degree polynomial L^1 regression to obtain *agnostic* learning algorithms for halfspaces under distribution assumptions, as well as extended these previously studied low-degree regression algorithms into the agnostic setting. Further work used low degree polynomial L^1 regression to obtain agnostic learning algorithms for many more problems, again under various distributional assumptions [OS06, BOW08, KOS08, GS10, Kan10, Wim10, HKM10, DHK$^+$10, CKKL12, ABL14, DSFT$^+$14, FV15, FK15, BCO$^+$15, CGG$^+$17, FKV17, DKK$^+$21].

Learning halfspaces. See the work of [DKK$^+$21] and references therein, for a historical discussion about the problem of learning halfspaces, as well as some up-to-date references regarding some problems connected to the one studied here.

Polynomial approximation theory. Polynomial approximation theory has been used extensively as a tool for studying halfspaces. Among other work, see [KKMS08, DGJ$^+$10, KLS09, Dan15, DKTZ20b, DKK$^+$21].

Other works in testing distributions. There is a large body of literature on finite sample guarantees for property testing of distributions. Algorithms developed within this framework are given samples of an input distribution and aim to distinguish the case in which the distribution has a specified property, from the case in which the distribution is far (in a reasonable distance metric) from any distribution with that property. Properties of interest include whether the distribution is uniform, independent, monotone, has high entropy or is supported by a large number of distinct elements. We mention a few specific results that are closest to the results in this chapter: Let p be a distribution on a discrete domain of size M. For a "known" distribution q (where the algorithm knows the value of q on every element of the domain, and does not need samples from it – e.g., when q is the uniform distribution), distinguishing whether p is the same as q from the case where p is ϵ-far (in L_1 norm) from q requires $\Theta(\sqrt{M}/\epsilon^2)$ samples [GR00, BFR$^+$00, BFF$^+$01, Pan08, DGPP16, DGK$^+$21]. For a more in depth discussion of the history and results in this area, see the monograph by Canonne [Can22].

Other frameworks of trusting agnostic learners. The work of Goldwasser, Rothblum, Shafer and Yehudayoff considers the question of how an untrusted prover can convince a learner that a hypothesis is approximately correct, and show that significantly less data is needed than that required for agnostic learning [GRSY20].

1.2 Preliminaries.

1.2.1 Standard definitions.

The definition of agnostic learning is as follows:

Definition 1. *An algorithm $\mathcal{A}$ is an* agnostic (ϵ, δ)-learning algorithm *for function class $\mathcal{F}$ relative to the distribution D, if given access to i.i.d. example-label pairs (x, y) distributed according to D_{pairs}, with the marginal distribution on the examples equal to D, the algorithm $\mathcal{A}$ with probability at least $1 - \delta$ outputs a circuit computing a function $\widehat{f}$, such that*

$$\Pr_{(x,y)\in_R D_{pairs}}[y \neq \widehat{f}(x)] \leq \min_{f\in\mathcal{F}} \left(\Pr_{(x,y)\in_R D_{pairs}}[f(x) \neq y]\right) + \epsilon.$$

The quantity $\Pr_{(x,y)\in_R D_{\text{pairs}}}[f(x) \neq y]$ is often called the *generalization error* of $\widehat{f}$ (a.k.a. *out-of-sample error* or *risk*).

The following is standard theorem about agnostic learning from ℓ_1-approximation. The proof is implicit in [KKMS08] and this theorem has been implicitly used in much subsequent work (see Subsection 1.1.4 for references). Let U be some domain we are working over.

Theorem 1. *Let $\{g_1, \cdots g_N\}$ be a collection of real-valued functions over U that can be evaluated in time T. Then, for every $\epsilon > 0$, there is a learning algorithm $\mathcal{A}$ for which the following is true. Let D be any distribution over U and let $\mathcal{F}$ be any class of Boolean functions over U, such that every element of $\mathcal{F}$ is ϵ-approximated in L^1 norm relative to the distribution D by some element of span $(g_1, \cdots, g_N)$. Then, $\mathcal{A}$ agnostically (ϵ, δ)-learns $\mathcal{F}$ relative to D. The algorithm $\mathcal{A}$ uses $\widetilde{O}\left(\frac{N}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right)$ samples and uses run-time polynomial in this number of samples and T.*

We will also need the definition of k-wise independent distributions:

Definition 2. *A distribution of a random variable $\boldsymbol{x}$ over $\{\pm 1\}^d$ is called k-wise independent (a.k.a. k-wise uniform) if for any size-k subset S of $\{1, \cdots, s\}$ the distribution of $\{x_i : i \in S\}$ is uniform over $\{\pm 1\}^k$.*

1.2.2 New framework: testing distributional assumptions of a learning algorithm.

Definition 3. *Let $\mathcal{A}$ be an agnostic (ϵ, δ_1)-learning algorithm for function class $\mathcal{F}$ relative to the distribution D. We say that an algorithm $\mathcal{T}$ is a* tester for the distributional assumption *of $\mathcal{A}$ if*

1. (Soundness) *Suppose a distribution D_{pairs} on example-label pairs is such that, given access to i.i.d. labeled examples from it, the algorithm $\mathcal{T}$ outputs "Yes" with probability at least $1/4$. Then $\mathcal{A}$, given access to i.i.d. labeled examples from the same distribution D_{pairs}, will with probability at least $1 - \delta_1$ output a circuit computing a function $\widehat{f}$, such that*

$$\Pr_{(x,y)\in_R D_{pairs}}[y \neq \widehat{f}(x)] \leq \min_{f\in\mathcal{F}} \left(\Pr_{(x,y)\in_R D_{pairs}}[f(x) \neq y]\right) + \epsilon.$$

2. (Completeness) *Suppose D_{pairs} is such that the marginal distribution on examples equals to D. Then, given i.i.d. example-label pairs from D_{pairs}, tester $\mathcal{T}$ outputs "Yes" with probability at least $3/4$.*

If this definition is satisfied, then we say that $(\mathcal{A},\mathcal{T})$ form a tester-learner pair.

Constants $1/4$ and $3/4$ in the definition above can without loss of generality be replaced with any other pair of constants $1-\delta_2$ and $1-\delta_3$ with $\delta_2 \in (0,1)$ and $\delta_3 \in (\delta_2, 1)$. See Appendix 1.9.1 for the proof via a standard repetition argument.

1.3 An efficient tester-learner pair for learning halfspaces.

We now describe our tester-learner pair for learning halfspaces under the Gaussian distribution. Roughly, the testing algorithm checks that the low-degree moments of the distribution on examples are close enough to those of the standard Gaussian distribution. The learning algorithm uses a low-degree polynomial regression. As explained earlier, both of the algorithms ignore examples whose absolute value is too high, which allows them to obtain accurate estimates of distribution moments.

Tester-learner pair for learning halfspaces:

- Let $C_1, \cdots, C_4$ be a collection of constants to be tuned appropriately. Define $s := 2\left\lfloor \frac{1}{2\epsilon^4} \ln^3\left(\frac{1}{\epsilon}\right)\right\rfloor$, $\Delta := \left\lfloor \frac{1}{\epsilon^4} \ln^4\left(\frac{1}{\epsilon}\right)\right\rfloor$, $t := C_1 \Delta \ln \Delta \sqrt{\log d} + \sqrt{2\ln\left(\frac{C_2 d}{\epsilon}\right)}$, $N_1 := \left\lceil d^{C_3 s}\right\rceil$ and $N_2 := \left\lceil t^{2\Delta} d^{C_4 \Delta}\right\rceil$.
- **Learning algorithm $\mathcal{A}$.** Given access to i.i.d. labeled samples $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \{\pm 1\}$ from an unknown distribution:
  1. Obtain N_1 many labeled samples $(\boldsymbol{x}_i, y_i)$.
  2. Discard all the samples $(\boldsymbol{x}_i, y_i)$ for which the absolute value of some coordinate $\left|(\boldsymbol{x}_i)_j\right|$ is greater than t.
  3. Run the algorithm of Theorem 1 on the remaining samples, with accuracy parameter $\frac{\epsilon}{10}$, allowed failure probability $\frac{1}{20}$, and taking the set of $\{g_i\}$ to be the set of monomials of degree at most s, i.e. the set $\left\{\prod_{j=1}^d x_j^{\alpha_j} : \sum_j \alpha_j \le s\right\}$. This gives us a circuit computing predictor $\widehat{f}$. Form a new predictor $\widehat{f'}$ that given $\boldsymbol{x}$ outputs (i) $\widehat{f}(\boldsymbol{x})$ if for all $j \in [d]$, the value of $\left|(\boldsymbol{x}_i)_j\right|$ is at most t. (ii) 1 if[8] for some $j \in [d]$, the value of $\left|(\boldsymbol{x}_i)_j\right|$ exceeds t.
- **Testing algorithm $\mathcal{T}$.** Given access to i.i.d. labeled samples $\boldsymbol{x} \in \mathbb{R}^d$ from an unknown distribution:

[8] This one's arbitrary. Can also output 0 in this case.

1. For each $j \in [d]$:
   (a) Estimate $\Pr[|x_j| > t]$ up to additive $\frac{\epsilon}{30n}$ with error probability $\frac{1}{100n}$.
   (b) If the estimate is at least $\frac{\epsilon}{10n}$, output **No** and terminate.
2. Draw N_2 fresh samples $\{\boldsymbol{x}_i\}$, and discard the ones for which the absolute value of some coordinate $\left|(\boldsymbol{x}_i)_j\right|$ is greater than t.
3. For every monomial $\prod_{j=1}^{d} x_j^{\alpha_j}$ of degree at most Δ, compute its empirical expectation w.r.t. the samples $\{\boldsymbol{x}_i\}$. If for any of them resulting value is not within $\frac{1}{2n^{\Delta}}$ of $\mathbb{E}_{\boldsymbol{z}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{j=1}^{d} x_j^{\alpha_j}\right] = \prod_{j=1}^{d}\left((\alpha_j-1)!!\cdot \mathbb{1}_{\alpha_j \text{ is even}}\right)$, output **No** and terminate.
4. Output **Yes.**

The following theorem shows that the above algorithms indeed satisfy the criteria for a tester-learner pair for learning halfspaces under the Gaussian distribution:

Theorem 2 (**Tester-learner pair for learning halfspaces under Gaussian distribution**). *Suppose the values $C_1, \cdots, C_4$ present in algorithms $\mathcal{A}$ and $\mathcal{T}$ are chosen to be sufficiently large absolute constants, also assume d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constant. Then, the algorithm $\mathcal{A}$ is an agnostic $(O(\epsilon), 0.1)$-learner for the function class of linear threshold functions over $\mathbb{R}^d$ under distribution $\mathbb{N}(0, I_{d\times d})$ and the algorithm $\mathcal{T}$ is an assumption tester for $\mathcal{A}$. The algorithms $\mathcal{A}$ and $\mathcal{T}$ both require only $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples and run-time. Additionally, The tester $\mathcal{T}$ is label-oblivious.*

Note that an $(O(\epsilon), 0.1)$-learner can be made an agnostic (ϵ, δ_1)-learner for any fixed constant δ_1 and still require only $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples and run-time via a standard repeat-and-check argument. The tester $\mathcal{T}$ for the original learner will remain an assumption tester for the new learner.

The proof of correctness of the above tester-learner pair for halfspaces makes use of the following lemmas, which will be proved in Section 1.5. Lemma 1 states that as long as the low-degree moments of a distribution are similar to the corresponding moments of the Gaussian distribution, then the distribution is concentrated and anti-concentrated when projected onto any direction. Lemma 2 states that as long as distribution D satisfies the "nice" properties of concentration and anti-concentration, then any halfspace can be approximated by a low-degree polynomial with respect to distribution D. Taken together, these lemmas will be used to show that for any distribution D, if the moments of D look similar to moments of the Gaussian distribution, then halfspaces are well-approximated by low degree polynomials under D.

Lemma 1 (**Low degree moment lemma for distributions**.). *Suppose D is a distribution over $\mathbb{R}^d$ and Δ is an even positive integer, such that for every monomial $\prod_{i=1}^{d} x_i^{\alpha_i}$ of degree at most Δ we have*

$$\left|\mathbb{E}_{\boldsymbol{x}\sim D}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right]\right| \leq \frac{1}{d^{\Delta}}.$$

Further, assume that $\Delta \geq \frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)$. Then, for every unit vector $\boldsymbol{v}$, the random variable $\boldsymbol{v}\cdot\boldsymbol{x}$ (with $\boldsymbol{x} \in_R D$) has the following properties

- ***Concentration:*** *For any even positive integer $s \leq \Delta$, we have* $\left(\mathbb{E}_{\boldsymbol{x} \in_R D}\left[|\boldsymbol{v} \cdot \boldsymbol{x}|^s\right]\right)^{1/s} \leq 2\sqrt{s}$.
- ***Anti-concentration:*** *for any real y, we have*

$$\Pr_{\boldsymbol{x} \in_R D}\left[\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y+\epsilon]\right] \leq O\left(\epsilon\right).$$

Lemma 2 (Low degree approximation lemma for halfspaces.). *Suppose D is a distribution on $\mathbb{R}^d$ and $\boldsymbol{v} \in \mathbb{R}^d$ is a unit vector, such that for some positive real parameters α, γ, ϵ and a positive integer parameter s_0 we have*

- ***Anti-concentration:*** *for any real y, we have* $\Pr_{\boldsymbol{x} \in_R D}\left[\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y+\epsilon]\right] \leq \alpha$,
- ***Concentration:*** $\left(\mathbb{E}_{\boldsymbol{x} \in_R D}\left[|\boldsymbol{v} \cdot \boldsymbol{x}|^{s_0}\right]\right)^{1/s_0} \leq \beta$, *for some $\beta \geq 1$.*

Also assume $s_0 > \frac{5\beta}{\epsilon^2}$ and that ϵ is smaller than some sufficiently small absolute constant. Then, for every $\theta \in \mathbb{R}$ and there is a polynomial $P(x)$ of degree at most $\frac{2\beta}{\epsilon^2} + 1$ such that

$$E_{\boldsymbol{x} \in_R D}\left[|P(\boldsymbol{v} \cdot \boldsymbol{x}) - \mathit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|\right] = O\left(\alpha + \epsilon + \frac{(8\beta)^{\frac{2\beta}{\epsilon^2}+1}}{2^{s_0}}\right).$$

Each coefficient of the polynomial P has magnitude of at most $O\left(2^{\frac{4\beta}{\epsilon^2}}\right)$.

1.4 Technical preliminaries.

1.4.1 Polynomial approximation theory.

We will need some standard facts about Chebychev polynomials and approximation of functions using them. See, for example, the text [Tre19] for comprehensive treatment of this topic. First, we define Chebychev polynomials and present relevant facts about them. On the interval $[-1, 1]$ the k-th Chebychev polynomial can be defined as[9] $T_k(x) := \cos\left(k \arccos(x)\right)$.

For any $k \geq 0$, the polynomial $T_k(x)$ maps $[-1, 1]$ to $[-1, 1]$ (this follows immediately from the definition). Also, it is known that the Chebyshev polynomials satisfy a recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),$$

with the first two polynomials being $T_0(x) = 1$ and $T_1(x) = x$.

To present a standard theorem from text [Tre19] about approximating functions with Chebyshev polynomials, we will need the standard notions of Lipschitz continuity and of bounded variation functions. A function f is said to be Lipschitz continuous on $[-1, 1]$ if there is some C

[9] One needs to check that $\cos(k\alpha)$ is indeed a polynomial in $\cos\alpha$, which follows by writing $\cos(k\alpha) = \frac{e^{ik\alpha}+e^{-ik\alpha}}{2} = \frac{1}{2}\left(\left(\cos\alpha + i\sin\alpha\right)^k + \left(\cos\alpha - i\sin\alpha\right)^k\right)$, expanding, observing that terms involving odd powers of $\sin\alpha$ cancel out, and using the identity $\sin^2\alpha = 1 - \cos^2\alpha$.

so for any $x, y \in [-1,1]$ we have that $|f(x) - f(y)| \leq C|x-y|$. For a differentiable function $f : [-w, w] \to \mathbb{R}$, the *total variation of* f is the L_1 norm of it's derivative, i.e.

$$\int_{-w}^{w} \left| \frac{df(x)}{dx} \right| dx.$$

If f has a single discontinuity at some point a and is differentiable everywhere else, then the total variation of f is defined as the sum of the following three terms (i) $\int_{-w}^{a} \left| \frac{df(x)}{dx} \right| dx$, (ii) the magnitude of the discontinuity at a and (iii) $\int_{a}^{w} \left| \frac{df(x)}{dx} \right| dx$. Analogously, the definition extends to functions that are differentiable outside of finitely many discontinuities[10]. We say "f is of bounded variation V" if the total variation of f is at most V.

We are now ready to state the following theorem about approximating functions using Chebyshev polynomials:

Theorem 3 (Consequence of Theorem 7.2 in the text [Tre19] (see also Theorem 3.1 on page 19 in the text [Tre19]))**.** *Let f be Lipschitz continuous on $[-1,1]$ and suppose the derivative f' is of bounded variation V. Define for $k \geq 0$*

$$a_k := \frac{1 + \mathbb{1}_{k>0}}{\pi} \int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}} \, dx.$$

Then, for any $s \geq 0$ we have

$$\max_{x \in [-1,1]} \left| f(x) - \sum_{k=0}^{s} a_k T_k(x) \right| = O\left(\frac{V}{s}\right).$$

The partial sums $\sum_{k=0}^{s} a_k T_k$ are called Chebyshev projections.

1.5 Proving the two main lemmas (Lemma 1, and Lemma 2) via polynomial approximation theory.

1.5.1 Propositions useful for proving both main lemmas.

Here we will present proposition that will be useful for proving both Lemma 1 and 2. We start with an observation that bounds the magnitude of the coefficients of Chebyshev polynomials.

Observation 1. *Let $f : \mathbb{R} \to [-1,1]$ be a Lipschitz continuous function. Let $s \geq 1$ be an integer, let $w \geq 1$ be a real number, and let $f_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w})$, where $a_k := \frac{1+\mathbb{1}_{k>0}}{\pi} \int_{-1}^{1} \frac{f(wy)T_k(y)}{\sqrt{1-y^2}} \, dy$. Then, the largest coefficient from among all the monomials of $f_s(x)$ has value of at most $O(s3^s)$.*

[10] It is also standard to consider more general functions, but we will not need that.

Proof. See Appendix 1.9.2. □

Proving both lemmas, we will be approximating certain functions using Chebyshev polynomials re-scaled to the window $[-w, w]$. The following proposition lets us bound the error between function f and its low-degree polynomial approximation, contributed by the region $(-\infty, w) \cup (w, +\infty)$.

Proposition 1. *Let f be a Lipschitz continuous function $\mathbb{R} \to [-1, 1]$. Let $s \geq 1$ be an integer and $w \geq 1$ be real-valued, and let $f_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w})$, where $a_k := \frac{1+\mathbb{1}_{k>0}}{\pi} \int_{-1}^{1} \frac{f(wy)T_k(y)}{\sqrt{1-y^2}}\, dy$. Then, for any distribution D, it is the case that*

$$\mathbb{E}_{x \in_R D}\left[|f(x) - f_s(x)|\, \mathbb{1}_{|x|>w}\right] \leq O\left(4^s \mathbb{E}_{x \in_R D}\left[|x|^s\, \mathbb{1}_{|x|>w}\right]\right).$$

Proof. See Appendix 1.9.3.

□

The following proposition, in turn, allows us to bound the expression we encounter in Proposition 1 in terms of a bound on the moments of distribution D.

Proposition 2. *Let D be a distribution on $\mathbb{R}$ and $s_0 \in \mathbb{Z}^{>0}$ such that*

$$\left(\mathbb{E}_{x \in_R D}\left[|x|^{s_0}\right]\right)^{1/s_0} \leq \beta.$$

Then, for any $k \in \mathbb{Z} \cap [0, s_0/2]$ and $w \in \mathbb{R}^+$ we have

$$\mathbb{E}_{x \in_R D}\left[|x|^k\, \mathbb{1}_{|x|>w}\right] \leq 2w^k \left(\frac{\beta}{w}\right)^{s_0}$$

Proof. See Appendix 1.9.4. □

1.5.2 Proof of low degree moment lemma for distributions(Lemma 1).

Let us recall the setting of Lemma 1. D is a distribution over $\mathbb{R}^d$ and Δ is an even positive integer, such that for every monomial $\prod_{i=1}^{d} x_i^{\alpha_i}$ of degree at most Δ we have

$$\left|\mathbb{E}_{\boldsymbol{x} \sim D}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right] - \mathbb{E}_{\boldsymbol{x} \sim \mathbb{N}(0, I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right]\right| \leq \frac{1}{d^{\Delta}}.$$

Further, we have that $\Delta \geq \frac{1}{\epsilon^4} \ln^4\left(\frac{1}{\epsilon}\right)$. Then, we would like to show that for every unit vector $\boldsymbol{v}$, the random variable $\boldsymbol{v} \cdot \boldsymbol{x}$ (with $\boldsymbol{x} \in_R D$) has the following properties

- **Concentration:** For any even integer $s \leq \Delta$, we have $\left(\mathbb{E}_{\boldsymbol{x} \in_R D}\left[|\boldsymbol{v} \cdot \boldsymbol{x}|^s\right]\right)^{1/s} \leq 2\sqrt{s}$.

- **Anti-concentration:** for any real-valued parameter $w \geq 1$, for any real y, we have

$$\Pr_{\boldsymbol{x} \in_R D}\left[\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y+\epsilon]\right] \leq O\left(\epsilon\right).$$

We start with the following observation saying that if moments of a distribution D are similar to standard Gaussian, then the expectation of a polynomial of a form $(\boldsymbol{v} \cdot \boldsymbol{x})^s$ for D is similar to the same expectation under standard Gaussian.

Observation 2. *Suppose D is a distribution over $\mathbb{R}^d$ and Δ is a positive integer, such that for every monomial $\prod_{i=1}^d x_i^{\alpha_i}$ of degree at most Δ we have $\left|\mathbb{E}_{\boldsymbol{x} \sim D}\left[\prod_{i=1}^d x_i^{\alpha_i}\right] - \mathbb{E}_{\boldsymbol{x} \sim \mathbb{N}(0,1)}\left[\prod_{i=1}^d x_i^{\alpha_i}\right]\right| \leq \frac{1}{d^\Delta}$. Then, for any unit vector $\boldsymbol{v}$ and integer $s \leq \Delta$ we have*

$$\left|\mathbb{E}_{\boldsymbol{x} \in_R D}\left[(\boldsymbol{v} \cdot \boldsymbol{x})^s\right] - \mathbb{E}_{\boldsymbol{x} \in_R \mathbb{N}(0, I_{d \times d})}\left[(\boldsymbol{v} \cdot \boldsymbol{x})^s\right]\right| \leq \frac{d^s}{d^\Delta}.$$

Proof. See Appendix 1.9.5. □

Let us now show the concentration property. Let s be even. Recall that for even s we have $\mathbb{E}_{\boldsymbol{x} \sim \mathbb{N}(0, I_{d \times d})}\left[(\boldsymbol{v} \cdot \boldsymbol{x})^s\right] = \mathbb{E}_{x' \sim \mathbb{N}(0,1)}\left[(x')^s\right] = (s-1)!! \leq s^{s/2}$. This, together with Observation 2 implies

$$\left(\mathbb{E}_{\boldsymbol{x} \sim D}\left[(\boldsymbol{v} \cdot \boldsymbol{x})^s\right]\right)^{1/s} \leq \left(s^{s/2} + \frac{d^s}{d^\Delta}\right)^{1/s} = \sqrt{s}\left(1 + \frac{d^{s-\Delta}}{s^{s/2}}\right)^{1/s} \leq 2\sqrt{s},$$

which is the *concentration* property we wanted to show.

Now, we proceed to the *anti-concentration* property. Recall that for this property we need to bound $\Pr_{\boldsymbol{x} \in_R D}\left[\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y+\epsilon]\right]$. To this end, we first approximate $\mathbb{1}_{z \in [y, y+\epsilon]}$ using the following function

$$g(z) := \begin{cases} 0 & \text{if } z \leq y - \epsilon, \\ \frac{z-(y-\epsilon)}{\epsilon} & \text{if } z \in [y-\epsilon, y], \\ 1 & \text{if } z \in [y, y+\epsilon], \\ \frac{(y+2\epsilon)-z}{\epsilon} & \text{if } z \in [y+\epsilon, y+2\epsilon], \\ 0 & \text{if } z \geq y + 2\epsilon. \end{cases} \tag{1.1}$$

The key properties of g are (i) $g(z) \geq \mathbb{1}_{z \in [y, y+\epsilon]}$ (ii) $g(z) \in [0,1]$ (ii) $g(z)$ is Lipschitz continuous (iii) the derivative $g'(z)$ is of bounded variation of $\frac{4}{\epsilon}$ (because the function has four discontinuities, each of magnitude $1/\epsilon$ and it stays constant in-between the discontinuities).

Let $w \geq 1$ be real-valued and s be an integer in $[1, \Delta/2]$, to be chosen later and let $g_s(x) := \sum_{k=0}^s a_k T_k(\frac{x}{w})$, where $a_k := \frac{1+\mathbb{1}_{k>0}}{\pi}\int_{-1}^1 \frac{g(wy)T_k(y)}{\sqrt{1-y^2}}\,dy$. Observation 3 and propositions 3 and 4 are

stated and proven below, and we use them no to get the following bound:

$$\Pr_{\boldsymbol{x}\in_R D}\left[\boldsymbol{v}\cdot\boldsymbol{x}\in[y,y+\epsilon]\right]\leq \mathbb{E}_{\boldsymbol{x}\in_R D}\left[g(\boldsymbol{v}\cdot\boldsymbol{x})\right]\leq$$

$$\overbrace{\mathbb{E}_{\boldsymbol{x}\in_R\mathbb{N}(0,I_{n\times n})}\left[g(\boldsymbol{v}\cdot\boldsymbol{x})\right]}^{O(\epsilon)\text{ by Observation 3}}+\overbrace{\mathbb{E}_{\boldsymbol{x}\in_R\mathbb{N}(0,I_{n\times n})}\left[\left|g_d(\boldsymbol{v}\cdot\boldsymbol{x})-g(\boldsymbol{v}\cdot\boldsymbol{x})\right|\right]}^{\mathrm{O}\left(4^s w^s\left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta}+\frac{w}{\epsilon s}\right)\text{ by Proposition 3}}+$$

$$+\overbrace{\left|\mathbb{E}_{\boldsymbol{x}\in_R D}\left[g_d(\boldsymbol{v}\cdot\boldsymbol{x})\right]-\mathbb{E}_{\boldsymbol{x}\in_R\mathbb{N}(0,I_{n\times n})}\left[g_d(\boldsymbol{v}\cdot\boldsymbol{x})\right]\right|}^{\mathrm{O}\left(4^s\frac{d^s}{d^\Delta}\right)\text{ by Proposition 4}}+$$

$$+\underbrace{\mathbb{E}_{\boldsymbol{x}\in_R D}\left[\left|g(\boldsymbol{v}\cdot\boldsymbol{x})-g_d(\boldsymbol{v}\cdot\boldsymbol{x})\right|\right]}_{\mathrm{O}\left(4^s w^s\left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta}+\frac{w}{\epsilon s}\right)\text{ by Proposition 3}}=O\left(\epsilon+4^d w^d\left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta}+\frac{w}{\epsilon d}+4^d\frac{n^d}{n^\Delta}\right).$$

Now, recall we assumed without loss of generality that $\Delta=\frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)$, so taking[11] $s=\frac{1}{10\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)$ and $w=\frac{10}{\epsilon^2}\ln^2\left(\frac{1}{\epsilon}\right)$ we get

$$\Pr_{\boldsymbol{x}\in_R D}\left[\boldsymbol{v}\cdot\boldsymbol{x}\in[y,y+\epsilon]\right]\leq O\left(\epsilon+4^d w^d\left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta}+\frac{w}{\epsilon d}+4^d\frac{n^d}{n^\Delta}\right)=$$

$$O\left(\epsilon+\left(\frac{40}{\epsilon^2}\ln^2\left(\frac{1}{\epsilon}\right)\right)^{\frac{1}{10\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}\left(\frac{1}{5}\right)^{\frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)}+4^{\frac{1}{10\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}\frac{1}{n^{\frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)-\frac{1}{10\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}}\right)=O(\epsilon).$$

The only thing left to do is to prove the observations referenced above.

Observation 3. *For the function g as defined in Equation 1.1, we have*

$$\mathbb{E}_{\boldsymbol{x}\in_R\mathbb{N}(0,I_{d\times d})}\left[g(\boldsymbol{v}\cdot\boldsymbol{x})\right]=O(\epsilon)$$

Proof. The function g has a range of $[0,1]$ and is supported on $[y-\epsilon,y+3\epsilon]$. Also, $\boldsymbol{v}\cdot\boldsymbol{x}$ is distributed as a standard one-dimensional Gaussian. Therefore, the probability that $\boldsymbol{v}\cdot\boldsymbol{x}$ lands in $[y-\epsilon,y+3\epsilon]$, is at most $O(\epsilon)$, which finishes the proof. □

Proposition 3. *Suppose D is a distribution over $\mathbb{R}^d$ and Δ is a positive integer, such that for every monomial $\prod_{i=1}^d x_i^{\alpha_i}$ of degree at most Δ we have $\left|\mathbb{E}_{\boldsymbol{x}\sim D}\left[\prod_{i=1}^d x_i^{\alpha_i}\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^d x_i^{\alpha_i}\right]\right|\leq\frac{1}{d^\Delta}$. Let s be an integer in $[1,\Delta/2]$, let $w\geq 1$ be a real-valued parameter, and suppose $g:[-w,w]\to[-1,1]$ is a Lipschitz function whose derivative g' is of Bounded variation V, and let*

[11] We also check that (taking ϵ small enough) s is indeed in $[1,\Delta/2]$, as was required earlier.

$g_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w})$, *where* $a_k := \frac{1+\mathbb{1}_{k>0}}{\pi} \int_{-1}^{1} \frac{g(wy)T_k(y)}{\sqrt{1-y^2}}\, dy$. *Then, it is the case that*

$$\mathbb{E}_{\boldsymbol{x}\in_R D}\left[|g(\boldsymbol{v}\cdot\boldsymbol{x}) - g_s(\boldsymbol{v}\cdot\boldsymbol{x})|\right] \leq O\left(4^s w^s \left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta} + \frac{Vw}{s}\right).$$

Proof. Proposition 1 and Proposition 2 imply

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[|g(\boldsymbol{v}\cdot\boldsymbol{x}) - g_d(\boldsymbol{v}\cdot\boldsymbol{x})|\, \mathbb{1}_{|\boldsymbol{v}\cdot\boldsymbol{x}|>w}\right] \leq$$
$$O\left(4^d \mathbb{E}_{\boldsymbol{x}\in_R D}\left[|\boldsymbol{v}\cdot\boldsymbol{x}|^d\, \mathbb{1}_{|\boldsymbol{v}\cdot\boldsymbol{x}|>w}\right]\right) \leq 4^d w^d \left(\frac{2\sqrt{\Delta}}{w}\right)^{\Delta} \frac{\Delta}{\Delta - d}.$$

To use Theorem 3, we need to bound the total variation of the function $\frac{\mathrm{d}g(wz)}{\mathrm{d}z} = wg'(wz)$. Inspecting the definition of total variation, we see that $g'(wz)$ has the same total variation as $g'(z)$, which is at most V. Therefore, the total variation of $\frac{\mathrm{d}g(wz)}{\mathrm{d}z}$ is at most Vw. Thus, we have by Theorem 3 that

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[|g(\boldsymbol{v}\cdot\boldsymbol{x}) - g_s(\boldsymbol{v}\cdot\boldsymbol{x})|\, \mathbb{1}_{|\boldsymbol{v}\cdot\boldsymbol{x}|\leq w}\right] \leq \max_{z\in[-w,w]} |g(z) - g_s(z)| \leq O\left(\frac{Vw}{s}\right).$$

Summing the two equations above and recalling that $s \leq \Delta/2$, our proposition follows. □

Proposition 4. *Suppose D is a distribution over $\mathbb{R}^d$ and Δ is a positive integer, such that for every monomial $\prod_{i=1}^d x_i^{\alpha_i}$ of degree at most Δ we have $\left|\mathbb{E}_{\boldsymbol{x}\sim D}\left[\prod_{i=1}^d x_i^{\alpha_i}\right] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,1)}\left[\prod_{i=1}^d x_i^{\alpha_i}\right]\right| \leq \frac{1}{d^\Delta}$. Let $g : \mathbb{R} \to [-1,1]$ be a Lipschitz continuous function, and $g_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w})$,where $a_k := \frac{1+\mathbb{1}_{k>0}}{\pi} \int_{-1}^{1} \frac{f(wy)T_k(y)}{\sqrt{1-y^2}}\, dy$. Then*

$$\left|\mathbb{E}_{\boldsymbol{x}\in_R D}\left[g_s(\boldsymbol{v}\cdot\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\in_R \mathbb{N}(0,I_{d\times d})}\left[g_s(\boldsymbol{v}\cdot\boldsymbol{x})\right]\right| = O\left(4^s \frac{d^s}{d^\Delta}\right).$$

Proof. Observation 1 implies that $g_s(z)$ is a degree s polynomial, whose largest coefficient is at most $s3^s$. Using Observation 2 for each of these monomials, we get

$$\left|\mathbb{E}_{\boldsymbol{x}\in_R D}\left[g_s\left(\boldsymbol{v}\cdot\boldsymbol{x}\right)\right] - \mathbb{E}_{\boldsymbol{x}\in_R \mathbb{N}(0,I_{d\times d})}\left[g_s(\boldsymbol{v}\cdot\boldsymbol{x})\right]\right| \leq O\left(s^2 3^s\right)\frac{d^s}{d^\Delta} = O\left(4^s \frac{d^s}{d^\Delta}\right).$$

□

1.5.3 Proof of low degree approximation lemma for halfspaces (Lemma 2).

Let us recall what we need to show to prove Lemma 2. Without loss of generality, we assume we are in one dimension. D is a distribution on $\mathbb{R}$, such that for some positive real parameters α, γ, ϵ and a positive integer parameter s_0 we have

- **Anti-concentration:** for any real y, we have $\Pr_{x\in_R D}[x \in [y, y+\epsilon]] \leq \alpha$,
- **Concentration:** $(\mathbb{E}_{x\in_R D}[|x|^{s_0}])^{1/s_0} \leq \beta$, for some $\beta \geq 1$.

Also we have $s_0 > \frac{5\beta}{\epsilon^2}$ and that ϵ is smaller than some sufficiently small absolute constant. Then, for every $\theta \in \mathbb{R}$ we would like to show there is a polynomial $P(x)$ of degree at most $\frac{2\beta}{\epsilon^2}+1$ such that

$$E_{x\in_R D}[|P(x) - \text{sign}(x-\theta)|] = O\left(\alpha + \epsilon + \frac{(8\beta)^{\frac{2\beta}{\epsilon^2}+1}}{2^{s_0}}\right).$$

Let $w > 1$ and $s \in Z^+$ be parameters, values of which will be set later. We will approximate the sign function with a polynomial in the following two steps:

- Approximate $\text{sign}(x-\theta)$ by a continuous function

$$f(x) := \begin{cases} 1 & \text{if } \frac{x-\theta}{\epsilon} > 1, \\ -1 & \text{if } \frac{x-\theta}{\epsilon} < -1, \\ \frac{x-\theta}{\epsilon} & \text{otherwise.} \end{cases}$$

- For a parameter s, approximate $f(x)$ by

$$f_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w}),$$

where

$$a_k := \frac{1+\mathbb{1}_{k>0}}{\pi}\int_{-1}^{1} \frac{f(wy)T_k(y)}{\sqrt{1-y^2}}\,dy.$$

First, we observe that f is a good approximator for $\text{sign}(x-\theta)$ with respect to D.

Proposition 5. *If D is a distribution over $\mathbb{R}$ such that for every $x_0 \in \mathbb{R}$ it holds that $\Pr_{x\in_R D}[x \in [x_0, x_0+\epsilon]] \leq \alpha$, then (with $f(x)$ defined as above) we have*

$$\mathbb{E}_{x\in_R D}[|f(x) - \textit{sign}(x-\theta)|] \leq 2\alpha.$$

Proof. The two functions differ only on $[\theta-\epsilon, \theta+\epsilon]$, with the absolute value of difference being at most 1. Since the distribution D cannot have probability mass more than 2α in this interval, the proposition follows. □

Secondly, we show that f_s is a good approximator to f with respect to D, within the region $[-w, w]$.

Proposition 6. *For any distribution D, we have*

$$\mathbb{E}_{x\in_R D}\left[|f(x) - f_s(x)|\,\mathbb{1}_{|x|\leq w}\right] \leq O\left(\frac{w}{\epsilon s}\right)$$

Proof. Using Theorem 3 we have

$$\mathbb{E}_{x\in_R D}\left[|f(x)-f_s(x)|\,\mathbb{1}_{|x|\le w}\right] \le \max_{x\in[-w,w]}|f(x)-f_s(x)| = \max_{y\in[-1,1]}|f(wy)-f_s(wy)| = O\left(\frac{w}{\epsilon s}\right).$$

□

Now, we put all the relevant propositions together to show the lemma. Using Propositions 1 and 2, we see that if we have $s \in \mathbb{Z}\cap[1,d_0/2]$ then

$$\mathbb{E}_{x\in_R D}\left[|f(x)-f_s(x)|\,\mathbb{1}_{|x|>w}\right] \le O\left(4^s\mathbb{E}_{x\in_R D}\left[|x|^s\,\mathbb{1}_{|x|>w}\right]\right) \le O\left(4^s 2w^s\left(\frac{\beta}{w}\right)^{s_0}\right)$$

Together with Proposition 6, this implies that

$$\mathbb{E}_{x\in_R D}\left[|f(x)-f_s(x)|\right] \le O\left(4^s 2w^s\left(\frac{\beta}{w}\right)^{s_0}\right) + O\left(\frac{w}{\epsilon s}\right)$$

This, in turn, together with Proposition 5 implies that

$$E_{x\in_R D}\left[|f_s(x)-\text{sign}(x-\theta)|\right] = O\left(\alpha+\frac{w}{\epsilon s}+4^s w^s\left(\frac{\beta}{w}\right)^{s_0}\right).$$

Taking[12] $w=2\beta$ and $s=\left\lceil\frac{2\beta}{\epsilon^2}\right\rceil$ we get

$$E_{x\in_R D}\left[|f_s(x)-\text{sign}(x-\theta)|\right] = O\left(\alpha+\epsilon+\frac{(8\beta)^{\left\lceil\frac{2\beta}{\epsilon^2}\right\rceil}}{2^{s_0}}\right) = O\left(\alpha+\epsilon+\frac{(8\beta)^{\frac{2\beta}{\epsilon^2}+1}}{2^{s_0}}\right).$$

Finally, we note that by Observation 1 we have that each coefficient of the polynomial f_s has a magnitude of at most $O(s3^s) = O\left(4^{\frac{2\beta}{\epsilon^2}}\right)$. This completes the proof of the low degree approximation lemma for halfspaces (Lemma 2).

1.6 Proof of Main Theorem via two main lemmas.

1.6.1 Truncated Gaussian has moments similar to Gaussian

Recall that our tester truncates the samples and checks that low-degree moments are close to the corresponding moments of a Gaussian. If the distribution is indeed Gaussian, the following proposition shows that this truncation step does not distort the moments too much.

[12]Recall that to do all this we needed that s is in $[1,s_0/2]$. Recall that by an assumption of the lemma we are proving we have $s_0 > \frac{5\beta}{\epsilon^2}$ and $\beta \ge 1$. Therefore, for ϵ smaller than some sufficiently small absolute constant we indeed have $\left\lceil\frac{2\beta}{\epsilon^2}\right\rceil \in [1,s_0/2]$.

Proposition 7. *Let $\prod_{i=1}^{d} x_i^{\alpha_i}$ be a monomial of degree at most Δ and t a real number in the set $\left[2\sqrt{\Delta}+1,+\infty\right)$. Then we have*

$$\left|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\middle|\forall i:\ |x_i|\leq t\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right]\right|\leq O\left(2^{\Delta}\Delta^{\frac{\Delta+2}{2}}t^{\Delta}e^{-\frac{t^2}{2}}\right).$$

Proof. If any of the α_i is odd, both expectations are zero, so the proposition follows trivially. So, without loss of generality, assume that each α_i is even. Also, without loss of generality, we can also assume that $d\geq\Delta$ and the α_i can be non-zero only for $i\in\{1,\cdots\Delta\}$. We prove the following observation separately:

Observation 4. *For $s\geq 0$, if $w\geq 2\sqrt{s}+1$, then it is the case that*

$$\mathbb{E}_{x\in_R\mathbb{N}(0,1)}\left[x^s\mathbb{1}_{|x|>w}\right]\leq O\left(w^s e^{-\frac{w^2}{2}}\right)$$

Proof. We have $\int_w^{+\infty} x^s e^{-\frac{x^2}{2}}\,dx=\int_w^{+\infty} e^{-\left(\frac{x^2}{2}-s\ln x\right)}\,dx$. For $x\geq w$, we have

$$\frac{d}{dx}\left(\frac{x^2}{2}-s\ln x\right)=x-\frac{s}{x}\geq w-\frac{s}{w},$$

which means

$$\left(\frac{x^2}{2}-s\ln x\right)\geq\frac{w^2}{2}-s\ln(w)+\left(w-\frac{s}{w}\right)(x-w).$$

Thus, we have

$$\int_w^{+\infty} x^s e^{-\frac{x^2}{2}}\,dx\leq e^{-\frac{w^2}{2}+s\ln(w)}\int_w^{+\infty} e^{-\left(w-\frac{s}{w}\right)(x-w)}\,dx=\frac{w^s e^{-\frac{w^2}{2}}}{\left(w-\frac{s}{w}\right)}\leq O\left(w^s e^{-\frac{w^2}{2}}\right).$$

□

Now, we consider the one-dimensional case of our proposition.

Observation 5. *Let s be a positive integer and t be a real number, such that t is in $[2\sqrt{s}+1,+\infty)$, then*

$$\left|\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^s\middle||x|\leq t\right]-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^s\right]\right|\leq O\left(t^s e^{-\frac{t^2}{2}}\right).$$

Proof. If s is odd, both expectations are zero, so without loss of generality assume that s is even.

We have

$$\left|\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\middle||x|\le t\right]-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\right]\right|=$$
$$\left|\frac{\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|\le t}\right]}{\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|\le t\right]}-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|\le t}\right]-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|>t}\right]\right|=$$
$$\left|\frac{\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|\le t}\right]\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|>t\right]}{\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|\le t\right]}-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|>t}\right]\right|\le$$

Using (i) triangle inequality (ii) $\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|\le t\right]\ge\Omega(1)$ because $t\ge 1$.

$$\overbrace{\le O\left(\left|\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|\le t}\right]\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|>t\right]\right|\right)+\left|\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|>t}\right]\right|}\le$$
$$\underbrace{\le O\left(d^{d/2}\left|\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|>t\right]\right|\right)+\left|\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^d\mathbb{1}_{|x|>t}\right]\right|}_{\text{Since }\mathbb{E}_{x\sim\mathbb{N}(0,1)}[x^s]=(s-1)!!.}\underbrace{\le O\left(d^{d/2}e^{-\frac{t^2}{2}}+t^de^{-\frac{t^2}{2}}\right)}_{\text{Using Observation 4.}}\underbrace{\le O\left(t^de^{-\frac{t^2}{2}}\right)}_{\text{Because }t>\sqrt{s}.}.$$

□

We proceed to reduce the high-dimensional case to the one-dimensional version we have just shown.

$$\left|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^n x_i^{\alpha_i}\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^n x_i^{\alpha_i}\middle|\forall i:\ |x_i|\le t\right]\right|=$$
$$\left|\prod_{i=1}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]-\prod_{i=1}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\middle||x|\le t\right]\right|\le$$
$$\sum_{j=1}^{\Delta}\left|\prod_{i=1}^{j-1}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\middle||x|\le t\right]\prod_{i=j}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]-\prod_{i=1}^{j}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\middle||x|\le t\right]\prod_{i=j+1}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]\right|=$$
$$\sum_{j=1}^{\Delta}\left|\prod_{i=1}^{j-1}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\middle||x|\le t\right]\prod_{i=j+1}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]\left(\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_j}\right]-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_j}\middle||x|\le t\right]\right)\right|.$$

Now, we have $\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\middle||x|\le t\right]=\frac{\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\mathbb{1}_{|x|\le t}\right]}{\Pr_{x\sim\mathbb{N}(0,1)}[|x|\le t]}\le 2\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]$, since $\Pr_{x\sim\mathbb{N}(0,1)}\left[|x|\le t\right]\ge 0.5$ for $t\ge 1$. Using this, Observation 5 and the fact that $\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]=(\alpha_j-1)!!\le\alpha_j^{\alpha_j/2}$ with

the inequality above, we have

$$\left|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^{n}x_i^{\alpha_i}\middle|\forall i:\ |x_i|\leq t\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^{n}x_i^{\alpha_i}\right]\right|\leq$$

$$2^{\Delta}\prod_{i=1}^{\Delta}\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_i}\right]\sum_{j=1}^{\Delta}\left|\left(\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_j}\right]-\mathbb{E}_{x\sim\mathbb{N}(0,1)}\left[x^{\alpha_j}\middle||x|\leq t\right]\right)\right|\leq$$

$$O\left(2^{\Delta}\prod_{j=1}^{\Delta}\alpha_j^{\alpha_j/2}\left(\sum_{j=1}^{\Delta}t^{\alpha_j}e^{-\frac{t^2}{2}}\right)\right)\leq O\left(2^{\Delta}\Delta^{\Delta/2}\left(\Delta\cdot t^{\Delta}e^{-\frac{t^2}{2}}\right)\right)=O\left(2^{\Delta}\Delta^{\frac{\Delta+2}{2}}t^{\Delta}e^{-\frac{t^2}{2}}\right)$$

This completes the proof of Proposition 7. □

1.6.2 Finishing the proof of Theorem 2.

In this subsection we finish the proof of Theorem 2, using the low degree moment lemma for distributions (Lemma 1) and the low degree approximation lemma for halfspaces (Lemma 2). The main thing left to do is to address issues relating to truncation of samples in the learning and testing algorithms.

We now restate the theorem. We are given that the values $C_1,\cdots,C_4$ present in algorithms $\mathcal{A}$ and $\mathcal{T}$ (in the beginning of Section section 1.3 on page 25) are chosen to be sufficiently large absolute constants, and also d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constant. Then, we need to show that the algorithm $\mathcal{A}$ is an agnostic $(O(\epsilon),0.1)$-learner for the function class of linear threshold functions over $\mathbb{R}^d$ under distribution $\mathbb{N}(0,I_{d\times d})$ and the algorithm $\mathcal{T}$ is an assumption tester for $\mathcal{A}$. We also need to show that $\mathcal{A}$ and $\mathcal{T}$ require only $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples and run-time.

Bounds on the run-time and sample complexity of our algorithms follow directly from our choice of parameters.

- The learner $\mathcal{A}$ draws $N_1:=d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples, then performs a computation running in time polynomial in (i) N_1 (ii) the number of monomials $\prod_{j=1}^{d}x_j^{\alpha_j}$ of degree at most s, which is $O\left(d^s\right)$ (this includes the run-time consumed by the algorithm of Theorem 1). Overall, the learner $\mathcal{A}$ uses $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples and run-time.
- The tester $\mathcal{T}$ first performs estimations of values $\Pr\left[|x_j|>t\right]$ up to additive $\frac{\epsilon}{30n}$ with error probability $\frac{1}{100n}$, which in total require $poly\left(\frac{d}{\epsilon}\right)$ samples and run-time. Then, the tester $\mathcal{T}$ obtains $N_2:=\left\lceil t^{\Delta}d^{C_4\Delta}\right\rceil$ samples (where $\Delta:=\left\lfloor\frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)\right\rfloor$ and $t:=C_1\Delta\ln\Delta\sqrt{\log d}+\sqrt{2\ln\left(\frac{C_2d}{\epsilon}\right)}$) and performs a polynomial time computation with them. We see that $t=O\left(poly\left(\frac{1}{\epsilon},d\right)\right)$ and therefore $N_2=d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$. Finally, the tester $\mathcal{T}$ runs a computation running in time polynomial in (i) N_2 and (ii) the number of monomials $\prod_{j=1}^{d}x_j^{\alpha_j}$ of degree at most Δ, which is $O\left(d^{\Delta}\right)$. Overall, we get that the run-time and sample complexity of $\mathcal{T}$ is $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$.

Proposition 8. *The following proposition uses the low degree approximation lemma for halfspaces (Lemma 2) to argue that, under certain regularity conditions on the distribution D, the learning algorithm satisfies the agnostic learning guarantee. Suppose the $C_1, \cdots, C_4$ are chosen to be sufficiently large absolute constants, d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constant. Suppose D is a distribution over $\mathbb{R}^d$ such that it the following properties hold*

- ***Good tail:*** *We have* $\mathrm{Pr}_{\boldsymbol{x}\in_R D}\left[\exists i \in [d] : \ |x_i| > t\right] \leq \frac{\epsilon}{5}$.
- **Concentration along any direction for truncated distribution:** For any unit vector $\boldsymbol{v}$ we have
$$\left(\mathbb{E}_{\boldsymbol{x}\in_R D}\left[|\boldsymbol{v}\cdot\boldsymbol{x}|^s \middle| \forall i \in [d] : \ |x_i| \leq t\right]\right)^{1/s} \leq 2\sqrt{s}.$$
- **A*nti-concentration along* any direction for truncated distribution*:*** For any unit vector $\boldsymbol{v}$ and for any real y, we have
$$\mathrm{Pr}_{\boldsymbol{x}\in_R D}\left[\boldsymbol{v}\cdot\boldsymbol{x} \in [y, y+\epsilon] \middle| \forall i \in [d] : \ |x_i| \leq t\right] \leq O\left(\epsilon\right).$$

Then, the algorithm $\mathcal{A}$ is an agnostic $\left(O\left(\epsilon\right), 0.1\right)$-learner for the function class of linear threshold functions over $\mathbb{R}^d$ under distribution D with failure probability at most $\frac{1}{20}$.

Proof. Let $D_{\text{truncated}}$ be the distribution of $\boldsymbol{x}$ drawn from D conditioned on $|x_i| \leq t$ for all i. We see that the premises of this proposition imply that the distribution $D_{\text{truncated}}$ satisfies the premises of the low degree approximation lemma for halfspaces(Lemma 2) with parameters $s_0 = s$, $\alpha = O(\epsilon)$ and $\beta = 2\sqrt{s}$. Taking ϵ smaller than some absolute constant ensures that the condition $s > \frac{5\beta}{\epsilon^2} = \frac{10\sqrt{s}}{\epsilon^2}$ is also satisfied.

The low degree approximation lemma for halfspaces(Lemma 2) then allows us to conclude that for every $\theta \in \mathbb{R}$ and for any $w \geq 1$ there is a polynomial $P(x)$ of degree at most s such that

$$E_{\boldsymbol{x}\in_R D_{\text{truncated}}}\left[|\text{sign}(\boldsymbol{v}\cdot\boldsymbol{x} - \theta) - P(\boldsymbol{v}\cdot\boldsymbol{x})|\right] = O\left(\epsilon + \frac{(16\sqrt{s})^{\frac{4\sqrt{s}}{\epsilon^2}+1}}{2^s}\right).$$

Recalling that $s := 2\left\lfloor \frac{1}{2\epsilon^4}\ln^3\left(\frac{1}{\epsilon}\right)\right\rfloor$ so we get that

$$E_{\boldsymbol{x}\in_R D_{\text{truncated}}}\left[|\text{sign}(\boldsymbol{v}\cdot\boldsymbol{x} - \theta) - P(\boldsymbol{v}\cdot\boldsymbol{x})|\right] = O\left(\epsilon + \frac{\left(O\left(\frac{1}{\epsilon^2}\ln^{1.5}\left(\frac{1}{\epsilon}\right)\right)\right)^{O\left(\frac{1}{\epsilon^4}\ln^{1.5}\left(\frac{1}{\epsilon}\right)\right)}}{2^{\Omega\left(\frac{1}{\epsilon^4}\ln^3\left(\frac{1}{\epsilon}\right)\right)}}\right).$$

For ϵ smaller than some sufficiently small absolute constant, the above is $O(\epsilon)$.

Thus, we have that for any linear threshold function $\text{sign}\left(\boldsymbol{v}\cdot\boldsymbol{x} - \theta\right)$ there is a degree s multivariate polynomial Q for which

$$\mathbb{E}_{\boldsymbol{x}\in_R D_{\text{truncated}}}\left[|\text{sign}\left(\boldsymbol{v}\cdot\boldsymbol{x} - \theta\right) - Q(\boldsymbol{x})|\right] \leq O(\epsilon)$$

In other words, under $D_{\text{truncated}}$, any linear threshold function sign $(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ is $O(\epsilon)$-approximated in L^1 by something in the span of set of monomials of degree at most s, i.e. the set $\left\{\prod_{j=1}^{d} x_j^{\alpha_j} : \sum_j \alpha_j \leq s\right\}$. Now, Theorem 1. tells us that with probability at least $1 - \frac{1}{20}$ the predictor $\widehat{f}$ given in step 3 has an error of at most $O(\epsilon)$ more than $\text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ for samples $\boldsymbol{x} \in_R D_{\text{truncated}}$. Overall, recalling the definition of $D_{\text{truncated}}$ we have

$$\begin{aligned}
&\Pr_{\boldsymbol{x}, y \in_R D_{\text{pairs}}}\left[\widehat{f'}(\boldsymbol{x}) \neq y\right] \leq \\
&\qquad \Pr_{\boldsymbol{x} \in_R D}\left[\exists i \in [n] : |x_i| > t\right] + \Pr_{\boldsymbol{x}, y \in_R D_{\text{pairs}}}\left[\widehat{f}(\boldsymbol{x}) \neq y \middle| \forall i \in [n] : |x_i| \leq t\right] \leq \\
&\qquad\qquad \Pr_{\boldsymbol{x}, y \in_R D_{\text{pairs}}}\left[\text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta) \neq y \middle| \forall i \in [n] : |x_i| \leq t\right] + O\left(\epsilon\right),
\end{aligned}$$

which completes the proof. □

Now, the following proposition, using low degree moment lemma for distributions (Lemma 1), tells us that the tester we use (1) is likely accept if the Gaussian assumption indeed holds (2) is likely to reject if the regularity conditions for Proposition 8 do not hold.

Proposition 9. *Suppose the $C_1, \cdots, C_4$ are chosen to be sufficiently large absolute constants, d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constant. Then, there is some absolute constant B, so the tester $\mathcal{T}$ has the following properties:*

1. *If $\mathcal{T}$ is given samples from $\mathbb{N}(0, I_{d \times d})$, it outputs **Yes** with probability at least 0.9.*
2. *The tester $\mathcal{T}$ rejects with probability greater than 0.9 any D for which at least one of the following holds:*
   (a) ***Bad tail:** We have $\Pr_{\boldsymbol{x} \in_R D}\left[\exists i \in [d] : |x_i| > t\right] > \frac{\epsilon}{5}$.*
   (b) ***Failure of concentration along some direction for truncated distribution:** there is a unit vector $\boldsymbol{v}$ such that*

$$\left(\mathbb{E}_{\boldsymbol{x} \in_R D}\left[|\boldsymbol{v} \cdot \boldsymbol{x}|^s \middle| \forall i \in [d] : |x_i| \leq t\right]\right)^{1/s} > 2\sqrt{s}.$$

   (c) ***Failure of anti-concentration along some direction for truncated distribution:** there is a unit vector $\boldsymbol{v}$ and real y, for which*

$$\Pr_{\boldsymbol{x} \in_R D}\left[\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y + \epsilon] \middle| \forall i \in [d] : |x_i| \leq t\right] > B\epsilon.$$

Proof. First, assume that $\mathcal{T}$ is getting samples from $\mathbb{N}(0, I_{d \times d})$ and let us prove that $\mathcal{T}$outputs ***Yes*** *with probability at least* 0.9.

Since $t \geq 1$, by we have[13] $\Pr_{z\in\mathbb{N}(0,1)}\left[|z| > t\right] \leq O\left(e^{-\frac{t^2}{2}}\right)$. As $t \geq \sqrt{2\ln\left(\frac{C_2 d}{\epsilon}\right)}$, taking C_2 large enough we get $\Pr_{z\in\mathbb{N}(0,1)}\left[|z| > t\right] \leq \frac{\epsilon}{30n}$. Therefore, $\mathbb{N}(0, I_{d\times d})$ passes step 1 of tester $\mathcal{T}$ with probability at least $1 - \frac{1}{100}$.

Also, $\Pr_{z\in\mathbb{N}(0,1)}\left[|z| > t\right] \leq \frac{\epsilon}{30n}$ implies that $\Pr_{\boldsymbol{x}\in\mathbb{N}(0,I_{d\times d})}\left[\forall i \in [d] : |x_i| \leq t\right] \geq 1 - \frac{\epsilon}{30}$. Together with a very loose application of the Hoeffding bound, we see that for sufficiently large C_4 with probability at least $1 - \frac{1}{100}$ only at most half of the samples are discarded in the step 2 of $\mathcal{T}$. We henceforth assume this indeed was the case. The remaining samples themselves are i.i.d. and distributed according to $\mathbb{N}(0, I_{d\times d})$ conditioned on all coordinates being in $[-t, t]$.

Since all remaining samples have the size of their coordinates bounded by t, the value of a given monomial $\prod_{j=1}^{d} x_j^{\alpha_j}$ of degree at most Δ evaluated on any of them is in $\left[-t^\Delta, t^\Delta\right]$. Therefore, the Hoeffding bound implies that for sufficiently large C_4 with probability at least $1 - \frac{1}{100n^\Delta}$ the empirical average of $\prod_{j=1}^{d} x_j^{\alpha_j}$ on the (at least $\frac{N_2}{2}$ many) remaining samples is within $\frac{1}{10n^\Delta}$ of

$$\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\middle|\forall i : |x_i| \leq t\right].$$

For sufficiently large C_1, we verify the premise of Proposition 7 that $t \in \left[2\sqrt{\Delta}+1, +\infty\right)$ and therefore have

$$\left|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\middle|\forall i : |x_i| \leq t\right] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right]\right| \leq O\left(2^\Delta \Delta^{\frac{\Delta+2}{2}} t^\Delta e^{-\frac{t^2}{2}}\right).$$

Now, we have $\frac{d}{dt}\left(\Delta\log t - \frac{t^2}{2}\right) = \frac{\Delta}{t} - t$ which is negative when $t > \sqrt{\Delta}$. As $t \geq C_1\Delta\left(\ln\Delta\sqrt{\log d}\right) > \sqrt{\Delta}$, we have

$$t^\Delta e^{-\frac{t^2}{2}} \leq \left(C_1\Delta\left(\ln\Delta\sqrt{\log d}\right)\right)^\Delta \exp\left(-\frac{\left(C_1\Delta\left(\ln\Delta\sqrt{\log d}\right)\right)^2}{2}\right),$$

which together with the preceding inequality implies

$$\left|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^{n} x_i^{\alpha_i}\middle|\forall i : |x_i| \leq t\right] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^{n} x_i^{\alpha_i}\right]\right| \leq$$
$$O\left(2^\Delta \Delta^{\frac{\Delta+2}{2}}\left(C_1\Delta\left(\ln\Delta\sqrt{\log n}\right)\right)^\Delta \exp\left(-\frac{\left(C_1\Delta\left(\ln\Delta\sqrt{\log n}\right)\right)^2}{2}\right)\right)$$

for sufficiently large C_1 the above is less than $\frac{1}{10n^\Delta}$. Therefore, in the whole, we have that the empirical average of $\prod_{j=1}^{d} x_j^{\alpha_j}$ in step 3 of $\mathcal{T}$ is with probability at least $1 - \frac{1}{100n^\Delta}$ within $\frac{1}{10n^\Delta}$ of

[13]Proof: $\int_t^{+\infty} e^{-\frac{x^2}{2}}\,dx \leq e^{-\frac{t^2}{2}}\int_t^{+\infty} e^{-\frac{t(x-t)}{2}}\,dx \leq \frac{2e^{-\frac{t^2}{2}}}{t} \leq O\left(e^{-\frac{t^2}{2}}\right)$.

$\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\right]$. Taking a union bound over all monomials $\prod_{j=1}^{d} x_j^{\alpha_j}$ of degree at most Δ, we see that the step 3 of the tester $\mathcal{T}$ also passes with probability at least $1-\frac{1}{100}$ when it is run on $\mathbb{N}(0, I_{d\times d})$.

Overall, we conclude that the probability $\mathcal{T}$ outputs **No** when given samples from $\mathbb{N}(0, I_{d\times d})$ is at most $\frac{3}{100} < 0.1$ as promised.

Now, we shall show that $\mathcal{T}$ will likely output **No** if any of the conditions given in the proposition hold.

If Condition (a) holds, we have $\Pr_{\boldsymbol{x}\in_R D}\left[\exists i \in [d] :\ |x_i| > t\right] > \frac{\epsilon}{5}$, then there is some coordinate i for which $\Pr_{\boldsymbol{x}\in_R D}\left[|x_i| > t\right] > \frac{\epsilon}{5n}$. This coordinate will lead to $\mathcal{T}$ outputting **No** in step 1 with probability at least $1-\frac{1}{100}$.

Now, suppose condition (a) doesn't hold so we $\Pr_{\boldsymbol{x}\in_R D}\left[\exists i \in [d] :\ |x_i| > t\right] \leq \frac{\epsilon}{5}$ but condition (b) or (c) does hold. We would like to show that $\mathcal{T}$ will still likely output **No.** With a very loose application of the Hoeffding bound, for sufficiently large C_4 with probability at least $1-\frac{1}{100}$ only at most half of the samples are discarded in the step 2 of $\mathcal{T}$, which we also assume henceforth. Using the Hoeffding bound again, we see that for sufficiently large C_4 with probability at least $1-\frac{1}{100}$ the empirical expectation of all monomials $\prod_{j=1}^{d} x_j^{\alpha_j}$ of degree at most Δ is within $\frac{1}{10n^{\Delta}}$ of

$$\mathbb{E}_{\boldsymbol{x}\in_R D}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\middle|\forall i \in [d] :\ |x_i| \leq t\right].$$

In other words, with probability at least $1-\frac{1}{100}$ the tester $\mathcal{T}$ will output **No** in step 3, unless we have for all monomials $\prod_{j=1}^{d} x_j^{\alpha_j}$ that

$$\left|\mathbb{E}_{\boldsymbol{x}\in_R D}\left[\prod_{i=1}^{d} x_i^{\alpha_i}\middle|\forall i \in [d] :\ |x_i| \leq t\right] - \mathbb{E}_{\boldsymbol{z}\sim\mathbb{N}(0,I_{d\times d})}\left[\prod_{j=1}^{d} x_j^{\alpha_j}\right]\right| \leq \frac{1}{2n^{\Delta}} + \frac{1}{10n^{\Delta}} = \frac{3}{5n^{\Delta}}.$$

So, to finish the proof, it is enough to show that the inequality above cannot hold if Condition (b) or Condition (c) holds. This follows from the low degree moment lemma for distributions(Lemma 1), for a sufficiently large choice of B, thereby finishing the proof[14]. □

Finally, we can use the two propositions above to finish the proof of Theorem 2. Bounds on run-time have been shown earlier, so now we need to show correctness. That requires us to show the following two conditions:

1. **(Soundness)** If, given access to i.i.d. labeled samples (x, y) distributed according to D_{pairs}, the algorithm $\mathcal{T}$ outputs "Yes" with probability at least $1/4$, then $\mathcal{A}$ will with probability at least 0.9 output a circuit computing a function $\widehat{f}$, such that

$$\Pr_{(x,y)\in_R D_{\text{pairs}}}[y \neq \widehat{f}(x)] \leq \min_{f\in\text{halfspaces}}\left(\Pr_{(x,y)\in_R D_{\text{pairs}}}[f(x) \neq y]\right) + O(\epsilon).$$

[14]To be explicit: if condition (a) doesn't hold but condition (b) or (c) does hold via union bound the probability that $\mathcal{T}$ will fail to output **No** is at most $\frac{1}{100}+\frac{1}{100} < 0.1$ as required.

2. **(Completeness)** Given access to i.i.d. labeled samples (x, y) distributed according to D_{pairs}, with x itself distributed as a Gaussian over R^d, tester $\mathcal{T}$ outputs "Yes" with probability at least $3/4$.

3. $\mathcal{A}$ is an agnostic learner for halfspaces over $\mathbb{R}^d$ under the Gaussian distribution.

Note that Condition 3 follows from the first two. The completeness condition (i.e. Condition 2) immediately follows from Proposition 9. The Soundness condition (i.e. Condition 1) follows from Proposition 9 and Proposition 8 in following way. If $\mathcal{T}$ outputs "No" with probability less than $3/4$ then conditions (a), (b) and (c) in Proposition 9 should all be violated. This allows us to use Proposition 8 to conclude that $\mathcal{A}$ is an agnostic $(O(\epsilon), 0.1)$-learner for the function class of linear threshold functions over $\mathbb{R}^d$ under distribution D, where D is the marginal distribution of x when (x, y) distributed according to D_{pairs}. This implies the Soundness condition (i.e. Condition 1 above) and finishes the proof of Theorem 2.

1.7 Tester-learner pairs for agnostically learning halfspaces under the uniform distribution over Boolean cube.

1.7.1 The tester-learner pair.

Tester-learner pair for learning halfspaces over $\{0,1\}^d$:

- Let C_1 be a sufficiently large constant to be tuned appropriately. Also define $k := \frac{1}{50\epsilon^4} \ln^4 \frac{1}{\epsilon}$.
- **Learning algorithm $\mathcal{A}_{\text{Boolean}}$.** Given access to i.i.d. labeled samples $(\boldsymbol{x}, y) \in \{\pm 1\}^d \times \{\pm 1\}$ from an unknown distribution:
  - Use the algorithm of Theorem 1 (that came from [KKMS08]), with error parameter $C_1\epsilon$, allowed failure probability $\frac{1}{10}$, and taking the set of $\{g_i\}$ to be the set of monomials of degree at most $\frac{20}{\epsilon^4} \ln^2 \frac{1}{\epsilon}$, i.e. the set $\left\{\prod_{j=1}^d x_j^{\alpha_j} : \sum_j \alpha_j \leq \frac{20}{\epsilon^4} \ln^2 \frac{1}{\epsilon}\right\}$ (with all $\alpha_j \in \{0,1\}$ because x_k are in $\{\pm 1\}$) .
- **Testing algorithm $\mathcal{T}_{\text{Boolean}}$.** Given access to i.i.d. labeled examples $\boldsymbol{x} \in \{\pm 1\}^d$ from an unknown distribution:
  1. Use a tester from literature (see [OZ18, AAK+07, AGM03]) for testing k-wise independent distributions against distributions that are $d^{-\frac{42}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}$-far from k-wise independent.
  2. Output the same response as the one given by the k-wise independence tester.

Theorem 4 (Tester-learner pair for learning halfspaces under uniform distribution on $\{\pm 1\}^d$). *Suppose the value C present in algorithm $\mathcal{A}_{\text{Boolean}}$ is chosen to be a sufficiently large absolute constant, also assume d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constants. Then, the*

algorithm $\mathcal{A}_{Boolean}$ is an agnostic $(O(\epsilon), 0.1)$-learner for the function class of linear threshold functions over $\{\pm 1\}^d$ under the uniform distribution and the algorithm $\mathcal{T}_{Boolean}$ is an assumption tester for $\mathcal{A}_{Boolean}$. The algorithms $\mathcal{A}_{Boolean}$ and $\mathcal{T}_{Boolean}$ both require only $d^{\tilde{O}\left(\frac{1}{\epsilon^4}\right)}$ samples and run-time. Additionally, the tester $\mathcal{T}_{Boolean}$ is label-oblivious.

The testers from the literature for k-wise independence take $d^{O(k)}/\eta^2$ samples and run-time to distinguish a k-wise independent distribution and a distribution that is η-far from k-wise independent (see [OZ18, AAK+07, AGM03]). Thus, the run-time of tester $\mathcal{T}_{\text{Boolean}}$ is $d^{\tilde{O}(1/\epsilon^4)}$. The same run-time bound of $d^{\tilde{O}(1/\epsilon^4)}$ for $\mathcal{A}_{\text{Boolean}}$ follows from Theorem 1.

The only thing remaining to prove is that the algorithm $\mathcal{A}_{\text{Boolean}}$ is indeed a $(O(\epsilon), 0.1)$-agnostic learning algorithm for the class of halfspaces on $\{\pm 1\}^d$ with respect to distributions D that are $d^{-\frac{42}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$-close to k-wise independent. By Theorem 1 (that came from [KKMS08]), this follows from the following proposition:

Proposition 17 (low-degree approximation)**.** *Let $sign(\boldsymbol{v}\cdot\boldsymbol{x}-\theta)$ be an arbitrary halfspace, $\boldsymbol{v}$ be normalized to be a unit vector, and let $k := \frac{1}{50\epsilon^4}\ln^4\frac{1}{\epsilon}$. Also let D be a distribution that is $d^{-\frac{42}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$-close in TV distance to k-wise independent. Then, there is a polynomial P of degree $\frac{20}{\epsilon^4}\ln^2\frac{1}{\epsilon}$ for which*

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[|P(\boldsymbol{x}) - sign(\boldsymbol{v}\cdot\boldsymbol{x}-\theta)|\right] = O(\epsilon)$$

The remaining subsections are dedicated to proving Proposition 17 which finishes the proof of Theorem 4.

1.7.2 Proving that halfspaces are well-approximated by low-degree polynomials under distributions close to k-wise independent.

Basic facts.

We now present some basic facts and definitions.

Definition 4 (From [DGJ+10])**.** *We say that the halfspace $sign(\boldsymbol{v}\cdot\boldsymbol{x}-\theta)$ is ϵ-regular if for any i we have $|v_i| \,/\, \|\boldsymbol{v}\| \le \epsilon$.*

The following is a standard corollary of the Berry-Esseen theorem (see for example Corollary 2.2 of [DGJ+10]).

Proposition 10. *Suppose the halfspace $sign(\boldsymbol{v}\cdot\boldsymbol{x}-\theta)$ is ϵ-regular, then for any interval $[a,b]\subset\mathbb{R}$ we have*

$$\Pr_{\boldsymbol{x}\sim\{\pm1\}^d}\left[\frac{\boldsymbol{v}\cdot\boldsymbol{x}}{\|\boldsymbol{v}\|}\in[a,b]\right] \le |b-a| + 2\epsilon.$$

We will also need the fact about the concentration properties of a k-wise independent distribution on $\{\pm1\}^d$, when it is projected to an arbitrary direction.

Proposition 11. *Suppose D is a k-wise independent distribution over $\{\pm 1\}^d$. Then, for any unit vector $\boldsymbol{v} \in \mathbb{R}^d$ and even integer $s \in [2, k]$, we have*

$$\left(\mathbb{E}_{\boldsymbol{x}\sim D}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^s\right]\right)^{1/d} \leq 2\sqrt{s},$$

Proof. Since $d \leq k$ and D is k-wise independent, we have

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^s\right] = \mathbb{E}_{\boldsymbol{x}\sim\{\pm1\}^d}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^s\right].$$

The standard Hoeffding bound tells us that for any $t \in \mathbb{R}$

$$\mathrm{Pr}_{\boldsymbol{x}\sim\{\pm1\}^d}\left[|\boldsymbol{v}\cdot\boldsymbol{x}| \geq t\right] \leq 2e^{-t^2/2}.$$

Therefore

$$\mathbb{E}_{\boldsymbol{x}\sim\{\pm1\}^n}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^d\right] = \mathbb{E}_{\boldsymbol{x}\sim\{\pm1\}^n}\left[\int_{\tau=0}^{\infty} d\tau^{d-1}\mathbb{1}_{|\boldsymbol{v}\cdot\boldsymbol{x}|>\tau}\,dt\right] =$$

$$= \underbrace{\int_0^\infty d\tau^{d-1}\mathrm{Pr}_{\boldsymbol{x}\sim\{\pm1\}^n}\left[|\boldsymbol{v}\cdot\boldsymbol{x}| > \tau\right]\,d\tau}_{\text{Via linearity of expectation and Tonelli's theorem.}} \leq 2\int_0^\infty d\tau^{d-1}e^{-\tau^2/2}\,d\tau =$$

$$\sqrt{2\pi}\cdot d\cdot\mathbb{E}_{\tau\sim N(0,1)}\left[|\tau|^{d-1}\right] = \sqrt{2\pi}\cdot d\cdot\sqrt{\frac{2}{\pi}}(d-2)!! = 2d!! \leq 2d^{d/2}.$$

This directly implies the statement we were seeking to prove. □

Re-using the polynomial from Section 1.5.2.

We will use the polynomial constructed in Section 1.5.2, which we designed to approximate well the function $\mathbb{1}_{[y,y+\epsilon]}$. We now summarize its properties

Proposition 12. *For every $y \in \mathbb{R}$, $\epsilon \in (0, 1]$, define*

$$g(z) := \begin{cases} 0 & \text{if } z \leq y-\epsilon, \\ \frac{z-(y-\epsilon)}{\epsilon} & \text{if } z \in [y-\epsilon, y], \\ 1 & \text{if } z \in [y, y+\epsilon], \\ \frac{(y+2\epsilon)-z}{\epsilon} & \text{if } z \in [y+\epsilon, y+2\epsilon], \\ 0 & \text{if } z \geq y+2\epsilon. \end{cases}$$

Then, for any $w \in \mathbb{R}_{>1}$, there exists a polynomial P_0 of degree $s = O(w/\epsilon^2)$, such that for any $x \in [-w, w]$ we have $|g(x) - P_0(x)| \leq \epsilon$. Additionally, each coefficient of P_0 has a magnitude of at most $s3^s$.

Proof of Proposition 17

First, we show that k-wise independent distributions are anti-concentrated when projected onto regular vectors.

Proposition 13. *Suppose the halfspace sign$(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ is ϵ-regular, $\boldsymbol{v}$ is normalized to be a unit vector, and let $k := \frac{1}{100\epsilon^4} \ln^4 \frac{1}{\epsilon}$. Then, for any k-wise independent distribution D we have for every $y \in \mathbb{R}$ that*

$$\Pr_{\boldsymbol{x} \sim D} [\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y + \epsilon]] = O(\epsilon)$$

Proof. We take $w := \frac{1}{\epsilon^2} \ln^2 \frac{1}{\epsilon}$, and WLOG assume that ϵ is small enough that $w > 1$. Let P_0 be as in Proposition 12. First, we would like to bound $\left|\mathbb{E}_{\boldsymbol{x} \sim D} \left[P_0 (\boldsymbol{v} \cdot \boldsymbol{x}) \mathbb{1}_{|\boldsymbol{v} \cdot \boldsymbol{x}| > w}\right]\right|$. To do this, first we observe that by combining Proposition 11 and Proposition 2 we have

$$\max_{i \in \{0, \cdots, s\}} \mathbb{E}_{\boldsymbol{x} \in_R D} \left[|\boldsymbol{v} \cdot \boldsymbol{x}|^i \mathbb{1}_{|x| > w}\right] \leq 2w^d \left(\frac{2\sqrt{k}}{w}\right)^k.$$

Each coefficient of P_0 is bounded by $s3^s$, this means that

$$|\mathbb{E}_{\boldsymbol{x} \sim D} [P_0 (\boldsymbol{v} \cdot \boldsymbol{x}) \mathbb{1}_{z > w}]| \leq 2s^2 3^s w^s \left(\frac{2\sqrt{k}}{w}\right)^k \leq O\left(4^s w^s \left(\frac{2\sqrt{k}}{w}\right)^k\right). \tag{1.2}$$

Repeating the exact same argument above for the uniform distribution over $\{\pm 1\}^d$ (in place of D) we also get

$$\left|\mathbb{E}_{\boldsymbol{x} \sim \{\pm 1\}^d} [P_0 (\boldsymbol{v} \cdot \boldsymbol{x}) \mathbb{1}_{z > w}]\right| \leq O\left(4^s w^s \left(\frac{2\sqrt{k}}{w}\right)^k\right). \tag{1.3}$$

Now, we consider the region inside $[-w, w]$. We have

$$5\epsilon \overbrace{\geq \Pr_{\boldsymbol{x} \sim \{\pm 1\}^d} [\boldsymbol{v} \cdot \boldsymbol{x} \in [y - \epsilon, y + 2\epsilon]]}^{\text{By Proposition 10.}} \overbrace{\geq \mathbb{E}_{\boldsymbol{x} \sim \{\pm 1\}^d} [P_0 (\boldsymbol{v} \cdot \boldsymbol{x}) \mathbb{1}_{z \leq w}] - \epsilon}^{\text{Because on } [-w, w] \text{ we have } P_0(z) \leq \mathbb{1}_{[y-\epsilon, y+2\epsilon]} + \epsilon} \tag{1.4}$$

Similarly, we also have

$$\mathbb{E}_{\boldsymbol{x} \sim D} [P_0 (\boldsymbol{v} \cdot \boldsymbol{x}) \mathbb{1}_{z \leq w}] \overbrace{\geq \Pr_{\boldsymbol{x} \sim D} [\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y + \epsilon]] - \epsilon}^{\text{Because on } [-w, w] \text{ we have } P_0(z) \geq \mathbb{1}_{[y, y+\epsilon]} - \epsilon}. \tag{1.5}$$

Taking together Equation (1.2), Equation (1.3), Equation (1.4) and Equation (1.5) we get

$$\Pr_{\boldsymbol{x} \sim D} [\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y + \epsilon]] \leq O(\epsilon) + O\left(4^s w^s \left(\frac{2\sqrt{k}}{w}\right)^k\right).$$

Substituting $k = \frac{1}{100\epsilon^4} \ln^4 \frac{1}{\epsilon}$, $s = O(w/\epsilon^2)$ and $w = \frac{1}{\epsilon^2} \ln^2 \frac{1}{\epsilon}$ we now get

$$\Pr_{\boldsymbol{x} \sim D} [\boldsymbol{v} \cdot \boldsymbol{x} \in [y, y + \epsilon]] \leq O(\epsilon) + O\left(4^s w^{s-k} \left(2\sqrt{k}\right)^k\right) = O(\epsilon)$$

□

Now, we use the proposition we just proved to show that, with respect to k-wise independent distributions, low-degree polynomials approximate well halfspaces whose normal vectors are regular.

Proposition 14. *Suppose the halfspace sign$(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ is ϵ-regular, $\boldsymbol{v}$ is normalized to be a unit vector, and let $k := \frac{1}{100\epsilon^4} \ln^4 \frac{1}{\epsilon}$. Then, for any k-wise independent distribution D we have a polynomial P of degree $s := \frac{1}{4\epsilon^4} \ln^2 \left(\frac{1}{\epsilon}\right)$ for which*

$$\mathbb{E}_{\boldsymbol{x} \sim D} [|P(\boldsymbol{x}) - \textit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|] = O(\epsilon)$$

Additionally, each coefficient of polynomial P is bounded by $(4n)^s$ in absolute value.

Proof. We combine Proposition 11 and Proposition 13 with Lemma 2. In Lemma 2, we have $\alpha = O(\epsilon)$, $s_0 = k$ and $\beta = 2\sqrt{s_0}$. Overall, from the conclusion of Lemma 2 it follows that for some polynomial $P(\boldsymbol{x}) = Q(\boldsymbol{v} \cdot \boldsymbol{x})$ it is indeed the case that

$$\mathbb{E}_{\boldsymbol{x} \sim D} [|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|] = O(\epsilon).$$

The degree of the polynomial P is $\frac{2\beta}{\epsilon^2} + 1$ which is at most $\frac{1}{4\epsilon^4} \ln^2 \left(\frac{1}{\epsilon}\right)$ for sufficiently small ϵ.

Now, we need to bound the (multivariable) coefficients of P. To do this, fix a specific multivariable term and track how much it can grow as we open the parentheses for $Q(\boldsymbol{v} \cdot \boldsymbol{x})$. As all coordinates of unit vector $\boldsymbol{v}$ are bounded by 1, every time we open the parentheses for a term of form $c_i(\boldsymbol{v} \cdot \boldsymbol{x})^i$, it can contribute at most $|c_i|\, d^i$ to the absolute value of any specific coefficient of P. As we know that every single-variable coefficient c_i of Q is bounded by $s3^s$, we get an overall bound of $s(3n)^s \leq (s4n)^s$ on each multivariate coefficient of P. □

Consequently, we use ideas similar to the ones in [DGJ$^+$10] in order to reduce the case of general halfspaces to the case of halfspaces whose normal vectors are regular.

Proposition 15. *Let sign$(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ be an arbitrary halfspace, $\boldsymbol{v}$ be normalized to be a unit vector, and let $k := \frac{1}{50\epsilon^4} \ln^4 \frac{1}{\epsilon}$. Then, for any k-wise independent distribution D we have a polynomial P of degree $\frac{20}{\epsilon^4} \ln^2 \frac{1}{\epsilon}$ for which*

$$\mathbb{E}_{\boldsymbol{x} \sim D} [|P(\boldsymbol{x}) - \textit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|] = O(\epsilon)$$

Additionally, each coefficient of the polynomial P has a magnitude of at most $d^{\frac{20}{\epsilon^4} \ln^2 \left(\frac{1}{\epsilon}\right)}$.

Proof. Without loss of generality, we assume that the values of $\boldsymbol{v}$ are in decreasing order (i.e. $v_i \geq v_{i+1}$). We use the notation $\sigma_i = \sqrt{\sum_{j>i} v_i^2}$. The **critical index** $\ell(\epsilon)$ is defined as the smallest i for which $v_i \leq \epsilon\sigma_i$. We set $\ell_0 = \frac{8\log^2(10/\epsilon)}{\epsilon^2}$ and consider two cases: (i) $\ell(\epsilon) \leq \ell_0$ and (ii) $\ell(\epsilon) > \ell_0$.

Suppose $\ell(\epsilon) \leq \ell_0$, then write the vector $\boldsymbol{v}$ as the concatenation of two vectors $\boldsymbol{v}_{\text{head}}$ in $\mathbb{R}^{\ell(\epsilon)}$ and $\boldsymbol{v}_{\text{head}}$ in $\mathbb{R}^{d-\ell(\epsilon)}$. Analogously a vector $\boldsymbol{x}$ in $\{\pm 1\}^d$ can be broken down into $\boldsymbol{x}_{\text{head}}$ in $\{\pm 1\}^{\ell(\epsilon)}$ and $\boldsymbol{x}_{\text{head}}$ in $\{\pm 1\}^{d-\ell(\epsilon)}$. For any fixed value of $\boldsymbol{x}_{\text{head}}$, the condition $\ell(\epsilon) \leq \ell_0$ directly implies that the halfspace $\text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_{\text{head}} + \boldsymbol{v}_{\text{tail}} \cdot \boldsymbol{x}_{\text{tail}} - \theta)$ is a regular halfspace. Since D is a k-wise independent distribution, when one conditions on a specific value of $\boldsymbol{x}_{\text{head}}$, the resulting distribution over $\boldsymbol{x}_{\text{tail}}$ is $k - \ell_0$-wise independent. Therefore by Proposition 14 there is some polynomial $P^{\boldsymbol{x}_{\text{head}}}(\boldsymbol{v}_{\text{tail}} \cdot \boldsymbol{x}_{\text{tail}})$ of degree $\frac{1}{4\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)$ for which we have:

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[\left|P^{\boldsymbol{x}_{\text{head}}}(\boldsymbol{v}_{\text{tail}} \cdot \boldsymbol{x}_{\text{tail}}) - \text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_{\text{head}} + \boldsymbol{v}_{\text{tail}} \cdot \boldsymbol{x}_{\text{tail}} - \theta)\right| \,\middle|\, \boldsymbol{x}_{\text{head}}\right] = O(\epsilon).$$

This means, that if we take our polynomial P to map $\boldsymbol{x} = (\boldsymbol{x}_{\text{head}}, \boldsymbol{x}_{\text{tail}})$ to $\sum_{\boldsymbol{x}_0 \in \{\pm 1\}^{\ell(\epsilon)}}(\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0} \cdot P^{\boldsymbol{x}_0}(\boldsymbol{x}_{\text{tail}}))$ then we will overall have:

$$\mathbb{E}_{\boldsymbol{x}\sim D}\left[|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v}\cdot\boldsymbol{x} - \theta)|\right] = O(\epsilon).$$

Since the indicators $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0}$ have degree of at most ℓ_0, the polynomial P has a degree of at most $\ell_0 + \frac{1}{4\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)$, which is at most $\frac{20}{\epsilon^4}\ln^2\frac{1}{\epsilon}$ for sufficiently small ϵ as required.

Let us bound the coefficients of P. For each fixed $\boldsymbol{x}_0$, we know that the coefficients of $P^{\boldsymbol{x}_0}(\boldsymbol{x}_{\text{tail}}))$ are bounded by $(4n)^{\frac{1}{4\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$. Each coefficient of $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0}$ is bounded by $\frac{1}{2^l}$ (this follows by explicitly writing out this polynomial). Overall, (since the variables in $P^{\boldsymbol{x}_0}(\boldsymbol{x}_{\text{tail}}))$ and $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0}$ are disjoint) we see that each coefficient of $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0} \cdot P^{\boldsymbol{x}_0}(\boldsymbol{x}_{\text{tail}})$ is bounded in absolute value by $(4n)^{\frac{1}{4\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}/2^l$. Summing this over all $\boldsymbol{x}_0$, we see that every coefficient of P is then at most $(4n)^{\frac{1}{4\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$ in absolute value. (This is at most $d^{\frac{20}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$ for sufficiently small ϵ).

This concludes our consideration of the case $\ell(\epsilon) \leq \ell_0$, and the rest of the proof examines the case $\ell(\epsilon) > \ell_0$.

Suppose we have $\ell(\epsilon) > \ell_0$. Similar to before, we break the vector $\boldsymbol{v}$ into $\boldsymbol{v}_{\text{head}}$ in $\mathbb{R}^{\ell_0}$ and $\boldsymbol{v}_{\text{tail}}$ in $\mathbb{R}^{d-\ell_0}$ and the vector $\boldsymbol{x}$ in $\{\pm 1\}^d$ into $\boldsymbol{x}_{\text{head}}$ in $\{\pm 1\}^{\text{tail}}$ and $\boldsymbol{x}_{\ell_0}$ in $\{\pm 1\}^{d-\ell_0}$. The polynomial we shall use to approximate the halfspace will now depend entirely on $\boldsymbol{x}_{\text{head}}$. Specifically, it will make the natural best guess at $\text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_{\text{head}} + \boldsymbol{v}_{\text{tail}} \cdot \boldsymbol{x}_{\text{tail}} - \theta)$ given only $\boldsymbol{x}_{\text{head}}$, i.e. we have P mapping $\boldsymbol{x} = (\boldsymbol{x}_{\text{head}}, \boldsymbol{x}_{\text{tail}})$ to $\sum_{\boldsymbol{x}_0 \in \{\pm 1\}^{\ell_0}}(\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0} \cdot \text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_0 - \theta))$. Since the indicators $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0}$ have degree of ℓ_0, the polynomial also has a degree of at most ℓ_0. Each of the indicators $\mathbb{1}_{\boldsymbol{x}_{\text{head}}=\boldsymbol{x}_0}$ has coefficients equal to $\frac{1}{2^{\ell_0}}$ in absolute value, and there at most d^{ℓ_0} of these indicator polynomials. Therefore, each coefficient of P is can be bounded by d^{ℓ_0} in absolute value.

We now want to argue that P has a small error. We will use the following proposition that is implicit in the proof of Theorem 5.4 of [DGJ+10].

Proposition 16. *For $\ell_0 = \frac{8\log^2(10/\epsilon)}{\epsilon^2}$, suppose D is a $(\ell_0 + 2)$-wise independent distribution over*

$\{\pm 1\}^d$, *sign*$(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ *is a halfspace with critical index* $\ell(\epsilon) > \ell_0$. *Also suppose* $\boldsymbol{v}$ *is a unit vector and its coordinates of* $\boldsymbol{v}$ *are in descending order, and break* $\boldsymbol{v}$ *into* $\boldsymbol{v}_{head}$ *in* $\mathbb{R}^{\ell_0}$ *and* $\boldsymbol{v}_{head}$ *in* $\mathbb{R}^{d-\ell_0}$ *and the vector* $\boldsymbol{x}$ *in* $\{\pm 1\}^d$ *into* $\boldsymbol{x}_{head}$ *in* $\{\pm 1\}^{\ell_0}$ *and* $\boldsymbol{x}_{\ell_0}$ *in* $\{\pm 1\}^{d-\ell_0}$. *Then we have*

$$\Pr_{\boldsymbol{x} \sim D}\left[\textit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta) \neq \textit{sign}(\boldsymbol{v}_{head} \cdot \boldsymbol{x}_{head} - \theta)\right] = O(\epsilon)$$

Now, when $\text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} \cdot \boldsymbol{x}_{\text{tail}} - \theta) = \text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_{\text{head}} - \theta)$ our polynomial has error zero, and when $\text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} \cdot \boldsymbol{x}_{\text{tail}} - \theta) \neq \text{sign}(\boldsymbol{v}_{\text{head}} \cdot \boldsymbol{x}_{\text{head}} - \theta)$ our polynomial has an error of 2. Overall, this means that indeed

$$\mathbb{E}_{\boldsymbol{x} \sim D}\left[|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|\right] = O(\epsilon).$$

□

Finally, we move from distributions that are k-wise independent to distributions that are merely close to k-wise independent, which concludes this line of reasoning.

Proposition 17 (low-degree approximation)**.** *Let* $\textit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)$ *be an arbitrary halfspace,* $\boldsymbol{v}$ *be normalized to be a unit vector, and let* $k := \frac{1}{50\epsilon^4} \ln^4 \frac{1}{\epsilon}$. *Also let* D *be a distribution that is* $d^{-\frac{42}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}$*-close in TV distance to* k*-wise independent. Then, there is a polynomial* P *of degree* $\frac{20}{\epsilon^4} \ln^2 \frac{1}{\epsilon}$ *for which*

$$\mathbb{E}_{\boldsymbol{x} \sim D}\left[|P(\boldsymbol{x}) - \textit{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|\right] = O(\epsilon)$$

Proof. Let D' be the closest in TV distance k-wise independent distribution to D. We have

$$d_{\text{TV}}(D, D') \leq d^{-\frac{42}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}.$$

By Proposition 15, we have a a polynomial P of degree $\frac{20}{\epsilon^4} \ln^2 \frac{1}{\epsilon}$ for which

$$\mathbb{E}_{\boldsymbol{x} \sim D'}\left[|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|\right] = O(\epsilon) \tag{1.6}$$

To move from D to D' we use the following observation that follows immediately from the definition of TV distance

Observation 6. *Let* φ *be some function* $\{\pm 1\}^d \to \mathbb{R}$ *and suppose* φ *is bounded everywhere by* B *in absolute value. Let* D *and* D' *be two probability distributions over* $\{\pm 1\}^d$. *Then*

$$|\mathbb{E}_{\boldsymbol{x} \sim D}[\varphi] - \mathbb{E}_{\boldsymbol{x} \sim D'}[\varphi]| \leq B \cdot d_{TV}(D, D')$$

The polynomial $P(\boldsymbol{x})$ has at most $d^{\frac{20}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}$ terms each of which has a coefficient of magnitude at most $d^{\frac{20}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}$. As each of the terms always evaluates to ± 1 anywhere on $\{\pm 1\}^d$, the absolute value of P is bounded by $d^{\frac{40}{\epsilon^4} \ln^2\left(\frac{1}{\epsilon}\right)}$. For all sufficiently small ϵ we therefore have that

$|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)| \leq d^{\frac{41}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)}$. This, together with the observation above gives us that

$$|\mathbb{E}_{\boldsymbol{x}\sim D}[|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|] - \mathbb{E}_{\boldsymbol{x}\sim D'}[|P(\boldsymbol{x}) - \text{sign}(\boldsymbol{v} \cdot \boldsymbol{x} - \theta)|]| \leq$$
$$n^{\frac{41}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)} d_{\text{TV}}(D, D') \leq n^{\frac{41}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)} n^{-\frac{42}{\epsilon^4}\ln^2\left(\frac{1}{\epsilon}\right)} = O(\epsilon)$$

Combining this with Equation (1.6) we finish the proof. □

1.8 Lower bounds on testable agnostic learning complexity.

In this section we present sample lower bounds for tester-learner pairs for (i) learning convex sets under Gaussian distribution in $\mathbb{R}^d$ (ii) learning monotone functions under uniform distribution over $\{0,1\}^d$.

1.8.1 Theorem statements.

The following theorem implies that there is no tester-learner pair for agnostic learning convex sets under the standard Gaussian distribution with combined sample complexity of $2^{o(d)}$.

Theorem 5. *For all sufficiently large d, the following is true. Suppose $\mathcal{A}$ is an algorithm that given sample-label pairs $\{(\boldsymbol{x_i}, y_i)\} \subset \mathbb{R}^d \times \{\pm 1\}$ outputs a function $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$. Also, suppose $\mathcal{T}$ is a tester that given access to i.i.d. labeled points $\{(\boldsymbol{x_i}, y_i)\} \subset \mathbb{R}^d \times \{\pm 1\}$ outputs "Yes" or "No". Suppose whenever the points $\{\boldsymbol{x_i}\}$ are themselves distributed i.i.d. from $\mathbb{N}(0, I_{d\times d})$, tester $\mathcal{T}$ outputs "Yes" with probability at least $1 - \delta_2$. Also suppose the combined sample complexity of $\mathcal{A}$ and $\mathcal{T}$ is at most $N := 2^{0.01n}$. Then, there is a distribution D_{pairs} on $\mathbb{R}^d \times \{\pm 1\}$ such that*

- *There is a function $f_0 : \mathbb{R}^d \to \{\pm 1\}$, for which $\{\boldsymbol{x} : f_0(x) = 1\}$ is a convex set and*

$$\Pr_{(\boldsymbol{x},y)\sim D_{pairs}}[f_0(\boldsymbol{x}) = y] = 1$$

  In other words, it predicts the label perfectly.

- *The tester $\mathcal{T}$, given samples from D_{pairs}, accepts with probability at least $1 - \delta_2 - \frac{1}{2^{\Omega(d)}}$.*

- *The learner $\mathcal{A}$, given samples from D_{pairs}, outputs a predictor $\widehat{f}$ whose expected advantage over random guessing is at most $\frac{1}{2^{\Omega(d)}}$.*

The following theorem implies that there is no tester-learner pair for agnostic learning monotone functions under uniform distribution over $\{0,1\}^d$ with combined sample complexity of $2^{o(d)}$. Recall that a function $f_0 : \{0,1\}^d \to \{\pm 1\}$ is *monotone* if $f_0(\boldsymbol{x}_1) \geq f_0(\boldsymbol{x}_2)$ whenever each coordinate of $\boldsymbol{x}_1$ is at least as large as the corresponding coordinate of $\boldsymbol{x}_2$.

Theorem 6. *For all sufficiently large d, the following is true. Suppose $\mathcal{A}$ is an algorithm that given sample-label pairs $\{(\boldsymbol{x_i}, y_i)\} \subset \{0,1\}^d \times \{\pm 1\}$ outputs a function $\widehat{f} : \{0,1\}^d \to \{\pm 1\}$. Also,*

suppose $\mathcal{T}$ is a tester that given access to i.i.d. labeled points $\{(\boldsymbol{x_i}, y_i)\} \subset \{0,1\}^d \times \{\pm 1\}$ outputs "Yes" or "No". Suppose whenever the points $\{\boldsymbol{x_i}\}$ are themselves distributed i.i.d. uniformly over $\{0,1\}^d$, tester $\mathcal{T}$ outputs "Yes" with probability at least $1-\delta_2$. Also suppose the combined sample complexity of $\mathcal{A}$ and $\mathcal{T}$ is at most $N := 2^{0.01n}$. Then, there is a distribution D_{pairs} on $\{0,1\}^d \times \{\pm 1\}$ such that

- *There is a monotone $f_0 : \{0,1\}^d \to \{\pm 1\}$ for which $\Pr_{(\boldsymbol{x},y)\sim D_{pairs}}[f_0(\boldsymbol{x}) = y] = 1$. In other words, it predicts the label perfectly.*
- *The tester $\mathcal{T}$, given samples from D_{pairs}, accepts with probability at least $1 - \delta_2 - \frac{1}{2^{\Omega(d)}}$.*
- *The learner $\mathcal{A}$, given samples from D_{pairs}, outputs a predictor $\widehat{f}$ whose expected advantage over random guessing is at most $\frac{1}{2^{\Omega\left(\frac{d}{\log^2 d}\right)}}$.*

1.8.2 Technical lemmas about behavior of testing and learning algorithms.

In this section we show lemmas that are helpful to show inability of testing and learning algorithms to perform well under certain circumstances. Roughly, the following lemma says that one can "fool" a tester for a specific distribution D by replacing it by a uniform sample from a set S of sufficiently large size, where each element in S is a uniform sample from D.

Lemma 3. *Let D be some fixed distribution over U. Suppose that a tester $\mathcal{T}$ outputs "Yes" with probability at least $1-\delta_2$ whenever given access to i.i.d. labeled samples $(x,y) \in U \times \{\pm 1\}$ distributed according to D_{pairs}, such that x itself is distributed according to D. Furthermore, suppose the number of samples consumed by $\mathcal{T}$ is at most N. Fix some function $g : U \to \{\pm 1\}$ and let S be a random multiset of M i.i.d. elements drawn from D. Then, with probability at least $1-\Delta$ over the choice of S we have*

$$\Pr_{\substack{x_1,\cdots,x_N\sim S \\ \text{randomness of } \mathcal{T}}}[\mathcal{T}((x_1,g(x_1)),\cdots,(x_N,g(x_N))) = \text{"Yes"}] \geq 1 - \delta_2 - \frac{N^2}{M} - \frac{N}{\sqrt{\Delta M}}.$$

Proof. Let the elements of the multiset S be $(z_1,\cdots,z_M)$, which recall are i.i.d. from D. Let $(z_{i_1},\cdots,z_{i_N})$ be sampled i.i.d. from S. We have

$$\begin{aligned}
&\Pr[\mathcal{T} \text{ accepts given } ((z_{i_1},g(z_{i_1})),\cdots,(z_{i_N},g(z_{i_N}))) \mid S = (z_1,\cdots,z_M)] \geq \\
&\Pr[\mathcal{T} \text{ accepts given } ((z_{i_1},g(z_{i_1})),\cdots,(z_{i_N},g(z_{i_N}))) \mid S = (z_1,\cdots,z_M), \forall j_1 \neq j_2 :\ i_{j_1} \neq i_{j_2}] \cdot \\
&\qquad \cdot \underbrace{\Pr[\forall j_1 \neq j_2 :\ i_{j_1} \neq i_{j_2}]}_{\geq 1-\frac{N^2}{M}, \text{ via birthay-paradox argument}} \geq \\
&\left(1 - \frac{N^2}{M}\right)\Pr[\mathcal{T} \text{ accepts given } ((z_{i_1},g(z_{i_1})),\cdots,(z_{i_N},g(z_{i_N}))) \mid S = (z_1,\cdots,z_M), \forall j_1 \neq j_2 :\ i_{j_1} \neq i_{j_2}]
\end{aligned}$$

In expectation, for the above probability we have

$$\Pr\left[\mathcal{T} \text{ accepts given } \left(\left(z_{i_1}, g\left(z_{i_1}\right)\right), \cdots, \left(z_{i_N}, g\left(z_{i_N}\right)\right)\right)\right] \geq$$
$$\geq \left(1 - \frac{N^2}{M}\right)\left(1 - \delta_2\right) \geq 1 - \delta_2 - \frac{N^2}{M},$$

because the conditioning on i_{j_1} and i_{j_2} being all distinct results in feeding $\mathcal{T}$ with i.i.d. uniform sample-label pairs for which we know the acceptance probability is at least $1 - \delta_2$, as given in the premise of the claim. Having bound the expectation of this probability, let us now bound its variance. Define

$$p_{\text{average}} :=$$
$$\mathbb{E}_S\left[\Pr\left[\mathcal{T} \text{ accepts given } \left(\left(z_{i_1}, g\left(z_{i_1}\right)\right), \cdots, \left(z_{i_N}, g\left(z_{i_N}\right)\right)\right) \mid S = \left(z_1, \cdots, z_M\right), \forall j_1 \neq j_2 : i_{j_1} \neq i_{j_2}\right]\right]$$

We have

$$\mathbb{E}_S\left[\left(\Pr\left[\mathcal{T} \text{ accepts given } \left(\left(z_{i_1}, g\left(z_{i_1}\right)\right), \cdots, \left(z_{i_N}, g\left(z_{i_N}\right)\right)\right) \mid S = \left(z_1, \cdots, z_M\right), \forall j_1 \neq j_2 : i_{j_1} \neq i_{j_2}\right] - p_{\text{average}}\right)^2\right] = \mathbb{E}\left[\left(\Pr\left[\mathcal{T} \text{ accepts given } \left\{\left(z_{k_1}, g\left(z_{k_1}\right)\right), \cdots, \left(z_{k_N}, g\left(z_{k_N}\right)\right)\right\}\right] - p_{\text{average}}\right)\right] \cdot$$
$$\cdot \left(\Pr\left[\mathcal{T} \text{ accepts given } \left\{\left(z_{l_1}, g\left(z_{l_1}\right)\right), \cdots, \left(z_{l_N}, g\left(z_{l_N}\right)\right)\right\}\right] - p_{\text{average}}\right)$$

where $\left\{k_1, \cdots, k_{N_{\text{tester}}}\right\}$ and $\left\{\ell_1, \cdots, \ell_{N_{\text{tester}}}\right\}$ are picked as i.i.d. uniform subsets of $\left\{1, \cdots, N_{\text{support}}\right\}$, with N_{tester} elements each.

Now, if it happens that $\left\{k_1, \cdots, k_{N_{\text{tester}}}\right\}$ and $\left\{\ell_1, \cdots, \ell_{N_{\text{tester}}}\right\}$ are disjoint, then $\left\{z_{k_1}, \cdots, z_{k_N}\right\}$ are independent from $\left\{z_{\ell_1}, \cdots, z_{\ell_N}\right\}$, and we check that the expectation above is then zero. Overall, this means that the expression above is upper-bounded by the probability that $\left\{k_1, \cdots, k_{N_{\text{tester}}}\right\}$ and $\left\{\ell_1, \cdots, \ell_{N_{\text{tester}}}\right\}$ have a non-zero intersection. Using a standard birthday-paradox argument, this is at most $\frac{N^2}{M}$.

Overall, over the choice of S, the quantity

$$\Pr\left[\mathcal{T} \text{ accepts given } \left(\left(z_{i_1}, g\left(z_{i_1}\right)\right), \cdots, \left(z_{i_N}, g\left(z_{i_N}\right)\right)\right) \mid S = \left(z_1, \cdots, z_M\right), \forall j_1 \neq j_2 : i_{j_1} \neq i_{j_2}\right]$$

has an expectation of at least $1 - \delta_2 - \frac{N^2}{M}$ and standard deviation of at most $\frac{N}{\sqrt{M}}$, so by Chebyshev's inequality it is at least $1 - \delta_2 - \frac{N^2}{M} - \frac{N}{\sqrt{\Delta M}}$ with probability at least $1 - \Delta$. This means that with probability at least $1 - \Delta$ we have

$$\Pr\left[\mathcal{T} \text{ accepts given } \left(\left(z_{i_1}, g\left(z_{i_1}\right)\right), \cdots, \left(z_{i_N}, g\left(z_{i_N}\right)\right)\right)\right] \geq 1 - \delta_2 - \frac{N^2}{M} - \frac{N}{\sqrt{\Delta M}}.$$

□

The following lemma says that if a function is "random enough", then a learning algorithm will not be able to get a non-trivially small error given few example-label pairs.

Lemma 4. *Let $\mathcal{A}$ be an algorithm that takes N samples $\{(x_i, y_i)\}$ with $x_i \in \{1, \cdots, M\}$ and $y_i \in \{\pm 1\}$ and outputs a predictor $\widehat{f} : \{1, \cdots, M\} \to \{\pm 1\}$. Let $g : \{1, \cdots, M\} \to \{\pm 1\}$ be a random function, such that (i) g has some predetermined (and possibly given to algorithm $\mathcal{A}$) values on some fixed subset of $\{1, \cdots, M\}$, which comprises an at most φ fraction of $\{1, \cdots, M\}$ (ii) g is i.i.d. uniformly random in $\{\pm 1\}$ on the rest of $\{1, \cdots, M\}$. Upon receiving N labeled samples $\{(x_i, g(x_i))\}$ with $\{x_i\}$ distributed i.i.d. uniformly on $\{1, \cdots, M\}$, let the algorithm $\mathcal{A}$ output a predictor $\widehat{f}$. Then, for sufficiently large M we have*

$$\mathbb{E}_{g,\, \{x_i\},\, \textit{randomness of } \mathcal{A}} \left[\left| \Pr_{x \in_R \{1, \cdots, M\}} \left[\widehat{f}(x) \neq g(x) \right] - \frac{1}{2} \right| \right] \leq \frac{3}{2} \left(\varphi + \frac{N}{M} \right) + 5 \sqrt{\frac{\ln M}{M}}.$$

Proof. Write $\{1, \cdots, M\}$ as a union of two disjoint sets S and $\overline{S}$, where S contains (i) the φM or fewer elements of $\{1, \cdots, M\}$ on which g is predetermined and (ii) the N or fewer elements of $\{1, \cdots, M\}$ that the learner $\mathcal{A}$ encountered among the labeled samples $\{(x_i, g(x_i))\}$. So, we have $|S| \leq N + \varphi M$. We can write

$$\Pr_{x \in_R \{1, \cdots, M\}} \left[\widehat{f}(x) \neq g(x) \right] = \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \in S} \right] + \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \notin S} \right],$$

which means

$$\begin{aligned}
&\left| \Pr_{x \in_R \{1, \cdots, M\}} \left[\widehat{f}(x) \neq g(x) \right] - \frac{1}{2} \right| \leq \\
&\quad \left| \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \in S} \right] + \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \in \overline{S}} \right] - \frac{|\overline{S}|}{2M} - \frac{|S|}{2M} \right| \leq \\
&\quad\quad \left| \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \notin S} \right] - \frac{|\overline{S}|}{2M} \right| + \frac{3\,|S|}{2M} \leq \\
&\quad\quad\quad \left| \mathbb{E}_{x \in_R \{1, \cdots, M\}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \mathbb{1}_{x \in \overline{S}} \right] - \frac{|\overline{S}|}{2M} \right| + \frac{3}{2} \left(\varphi + \frac{N}{M} \right) = \\
&\quad\quad\quad\quad \frac{|\overline{S}|}{M} \left| \mathbb{E}_{x \in_R \overline{S}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \right] - \frac{1}{2} \right| + \frac{3}{2} \left(\varphi + \frac{N}{M} \right)
\end{aligned}$$

Note that $\widehat{f}$ depends only on (i) S, (ii) values of g on S and (iii) the internal randomness of $\mathcal{A}$. This means that even conditioned on $\widehat{f}(x)$, the values of g on $\overline{S}$ are i.i.d. In other words, $\mathbb{E}_{x \in_R \overline{S}} \left[\mathbb{1}_{\widehat{f}(x) \neq g(x)} \right]$ is distributed as the average of $|\overline{S}|$ i.i.d. random variables, each of which is

uniformly random in $\{0,1\}$. A Hoeffding bound argument then implies that for any $\epsilon \in [0,1]$

$$\mathbb{E}_{g,\,\{x_i\},\,\text{randomness of }\mathcal{A}}\left[\left|\mathbb{E}_{x\in_R\overline{S}}\left[\mathbb{1}_{\widehat{f}(x)\neq g(x)}\right]-\frac{1}{2}\right|\right]\leq \epsilon + 2e^{-2\epsilon^2\left|\overline{S}\right|},$$

and taking $\epsilon=\sqrt{\frac{\ln\left|\overline{S}\right|}{2\left|\overline{S}\right|}}$,we get

$$\mathbb{E}_{g,\,\{x_i\},\,\text{randomness of }\mathcal{A}}\left[\left|\mathbb{E}_{x\in_R\overline{S}}\left[\mathbb{1}_{\widehat{f}(x)\neq g(x)}\right]-\frac{1}{2}\right|\right]\leq \underbrace{\sqrt{\frac{\ln\left|\overline{S}\right|}{2\left|\overline{S}\right|}}+\frac{2}{\left|\overline{S}\right|}\leq 5\sqrt{\frac{\ln M}{M}}}_{\text{Since } M\geq\left|\overline{S}\right|\geq M/2}.$$

Overall, we get

$$\mathbb{E}_{g,\,\{x_i\},\,\text{randomness of }\mathcal{A}}\left[\left|\Pr_{x\in_R\{1,\cdots,M\}}\left[\widehat{f}(x)\neq g(x)\right]-\frac{1}{2}\right|\right]\leq 5\sqrt{\frac{\ln M}{M}}+\frac{3}{2}\left(\varphi+\frac{N}{M}\right)$$

□

1.8.3 Propositions to be used in proving Theorem 5.

We will need a result about concentration the norm of an d-dimensional standard Gaussian. Roughly speaking, the norm is tightly concentrated within a $O\left(d^{1/4}\right)$-neighborhood of $\sqrt{d}$. More precisely, we use the following special case of Lemma 8.1 in [Bir01] (this reference contains a complete short proof):

Lemma 5. *Let $\boldsymbol{X}$ be a standard d-dimensional Gaussian, then for any $\alpha>0$ we have*

$$\Pr\left[|\boldsymbol{X}|^2\geq d+2\sqrt{d\ln\left(\frac{2}{\alpha}\right)}+2\ln\left(\frac{2}{\alpha}\right)\right]\leq\frac{\alpha}{2},$$

and

$$\Pr\left[|\boldsymbol{X}|^2\leq d-2\sqrt{d\ln\left(\frac{2}{\alpha}\right)}\right]\leq\frac{\alpha}{2}.$$

The following claim tells us that two independent Gaussian vectors are unlikely to be very close to each other.

Claim 1. *Let $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ be i.i.d. d-dimensional standard Gaussians. For all sufficiently large d, and for any $r>0$ we have*

$$\Pr\left[|\boldsymbol{X}_1-\boldsymbol{X}_2|\leq r\right]\leq 8^d\left(\frac{r^2}{d}\right)^{d/2}.$$

Proof. Probability density of a Gaussian is everywhere at most $\left(\frac{1}{\sqrt{2\pi}}\right)^d$, and the volume of a ball around $\boldsymbol{X}_1$ of radius r is $\frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}r^d$. Stirling's approximation formula tells that for sufficiently large d we have $\Gamma\left(\frac{d}{2}+1\right) \geq \sqrt{d}\left(\frac{d}{2e}\right)^{d/2}$. Therefore, for sufficiently large d

$$\frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}r^d \leq \frac{1}{\sqrt{d}}\left(2e\pi\right)^{d/2}\left(\frac{r^2}{d}\right)^{d/2} \leq 18^d\left(\frac{r^2}{d}\right)^{d/2}.$$

Overall, the probability that $|\boldsymbol{X}_2 - \boldsymbol{X}_1| \leq r$ is then at most $\left(\frac{1}{\sqrt{2\pi}}\right)^d 18^d\left(\frac{r^2}{d}\right)^{d/2} \leq 8^d\left(\frac{r^2}{d}\right)^{d/2}$, which finishes the proof. □

We will also need the following geometric observations for proving Theorem 5. In the following, we will use conv $(\cdot, \cdots, \cdot)$ to denote the convex hull of some number of objects. We will also use $\mathcal{B}_r$ to denote the ball $\{x : |x| \leq r\}$ in $\mathbb{R}^d$.

Claim 2. *Let $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ be points in $\mathbb{R}^d$ satisfying $|\boldsymbol{X}_1|, |\boldsymbol{X}_2| \in [a, b]$ for some $a > 0$ and $b > a$. Then, we have that if $|\boldsymbol{X}_2 - \boldsymbol{X}_1|$ is greater than $2\sqrt{b^2 - a^2}$, then the line segment connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ intersects $\mathcal{B}_a$.*

Proof. We show the claim by arguing that if $|\boldsymbol{X}_1|, |\boldsymbol{X}_2| \in [a, b]$ and the distance between the line segment connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and origin is at least a, then $|\boldsymbol{X}_2 - \boldsymbol{X}_1|$ is at most $2\sqrt{b^2 - a^2}$. If $|\boldsymbol{X}_1| \notin \{a, b\}$, then one can add a small multiple of $\boldsymbol{X}_1 - \boldsymbol{X}_2$ to $\boldsymbol{X}_1$ and this will increase the distance $|\boldsymbol{X}_2 - \boldsymbol{X}_1|$, while keeping the conditions satisfied. If $|\boldsymbol{X}_2| \notin \{a, b\}$, analogous argument applies. Therefore, without loss of generality $|\boldsymbol{X}_1|, |\boldsymbol{X}_2| \in \{a, b\}$. If both $|\boldsymbol{X}_1|$ and $|\boldsymbol{X}_2|$ equal to a, the segment will get closer than a to origin, unless $\boldsymbol{X}_1 = \boldsymbol{X}_2$ and $|\boldsymbol{X}_2 - \boldsymbol{X}_1| = 0$. If both $|\boldsymbol{X}_1|$ and $|\boldsymbol{X}_2|$ equal to b, then their distance is at most $2\sqrt{b^2 - a^2}$. Finally, we need to consider the case $|\boldsymbol{X}_1| = a$ and $|\boldsymbol{X}_2| = b$ (the case $|\boldsymbol{X}_1| = b$ and $|\boldsymbol{X}_2| = a$ is analogous). If $\boldsymbol{X}_1 \cdot (\boldsymbol{X}_2 - \boldsymbol{X}_1) < 0$, then for any sufficiently small κ we have $|\boldsymbol{X}_1 + \kappa(\boldsymbol{X}_2 - \boldsymbol{X}_1)|^2 = |\boldsymbol{X}_1|^2 + \kappa \boldsymbol{X}_1 \cdot (\boldsymbol{X}_2 - \boldsymbol{X}_1) + \kappa^2 |(\boldsymbol{X}_2 - \boldsymbol{X}_1)| < |\boldsymbol{X}_1|^2$, which means that sist(line segment connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, origin) $< a$ contradicting one of the conditions. Therefore, $\boldsymbol{X}_1 \cdot (\boldsymbol{X}_2 - \boldsymbol{X}_1) \geq 0$. We have

$$b^2 = |\boldsymbol{X}_2|^2 = |\boldsymbol{X}_1 + (\boldsymbol{X}_2 - \boldsymbol{X}_1)|^2 = |\boldsymbol{X}_1|^2 + |\boldsymbol{X}_2 - \boldsymbol{X}_1|^2 + \boldsymbol{X}_1 \cdot (\boldsymbol{X}_2 - \boldsymbol{X}_1) \geq a^2 + |\boldsymbol{X}_2 - \boldsymbol{X}_1|^2.$$

Therefore, $|\boldsymbol{X}_2 - \boldsymbol{X}_1| \leq \sqrt{b^2 - a^2}$ in this case. Overall across the cases, $|\boldsymbol{X}_2 - \boldsymbol{X}_1|$is at most $2\sqrt{b^2 - a^2}$. □

The following claim says that if the line segment between two points x_1 and x_2 intersects the ball $\mathcal{B}_a$, then (i) the convex hull of x_1 and $\mathcal{B}_a$ (ii) the convex hull of x_2 and $\mathcal{B}_a$ have no non-trivial intersection.

Claim 3. *For any $a > 0$, let x_1 and x_2 be points in $\mathbb{R}^n$ and suppose $x_1, x_2 \notin \mathcal{B}_a$. Then, if the line segment between x_1 and x_2 intersects $\mathcal{B}_a$, then conv $(x_{i_1}, \mathcal{B}_a) \cap$ conv $(x_{i_2}, \mathcal{B}_a) = \mathcal{B}_a$.*

Proof. We argue that $\text{conv}\,(\boldsymbol{X}_1, \mathcal{B}_a) \cap \text{conv}\,(\boldsymbol{X}_2, \mathcal{B}_a) \neq \mathcal{B}_a$ implies that the distance between the line segment connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and origin is greater than a. Indeed, let Z be a point in $\text{conv}\,(\boldsymbol{X}_1, \mathcal{B}_a) \cap \text{conv}\,(\boldsymbol{X}_2, \mathcal{B}_a)$ and not in $\mathcal{B}_a$. Then, since $\mathcal{B}_a$ is convex, the separating hyperplane theorem tells us that there is a hyperplane separating Z from $\mathcal{B}_a$. Now, $\boldsymbol{X}_1$ cannot be on the same side of the hyperplane as $\mathcal{B}_a$, because this would mean that the hyperplane separates Z from $\text{conv}\,(\boldsymbol{X}_1, \mathcal{B}_a)$. So, $\boldsymbol{X}_1$ has to be on the same side of the hyperplane as Z or be on the hyperplane itself. The same argument tells us that $\boldsymbol{X}_2$ has to be on the same side of the hyperplane as Z or be on the hyperplane itself. Overall, $\mathcal{B}_a$ is on one side of the hyperplane while any point on line segment connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is either on the other side or on the hyperplane itself. Since $\mathcal{B}_a$ is closed, the distance between $\mathcal{B}_a$ and the hyperplane is positive. This means $\text{sist}(\text{line segment connecting } \boldsymbol{X}_1 \text{ and } \boldsymbol{X}_2, \text{origin}) > a$. □

Claim 4. *For any $a > 0$, let $\{\boldsymbol{x}_i\}_{i=1}^M$ be a collection of points in $\mathbb{R}^d$ and suppose $\boldsymbol{x}_i \notin \mathcal{B}_a$ for all i. Also, suppose that for any distinct i_1 and i_2 the line segment between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ intersects $\mathcal{B}_a$. Then,*

$$\textit{conv}\,(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M, \mathcal{B}_a) = \textit{conv}\,(\boldsymbol{x}_1, \mathcal{B}_a) \cup \textit{conv}\,(\boldsymbol{x}_2, \mathcal{B}_a) \cup \cdots \cup \textit{conv}\,(\boldsymbol{x}_M, \mathcal{B}_a)\,.$$

Proof. The line segment from $\boldsymbol{x}_{i_1}$ to $\boldsymbol{x}_{i_2}$ can be decomposed into three contiguous nonempty disjoint regions, (i) the one in $\text{conv}\,(\boldsymbol{x}_{i_1}, \mathcal{B}_a) \setminus \mathcal{B}_a$ (ii) the one in $\mathcal{B}_a$ (iii) the one in $\text{conv}\,(\boldsymbol{x}_{i_2}, \mathcal{B}_a) \setminus \mathcal{B}_a$. This implies the following. Let $\beta\boldsymbol{x}_{i_1} + (1-\beta)\boldsymbol{x}_{i_2}$, with β in $[0,1]$, be an element of this line segment. If $\beta\boldsymbol{x}_{i_1} + (1-\beta)\boldsymbol{x}_{i_2}$ is in regions (i) or (ii) then we can write $\beta\boldsymbol{x}_{i_1} + (1-\beta)\boldsymbol{x}_{i_2} = \gamma\boldsymbol{x}_{i_1} + (1-\gamma)\boldsymbol{q}$ for some $\gamma \in [0,1]$ and some $\boldsymbol{q} \in \mathcal{B}_a$. If $\beta\boldsymbol{x}_{i_1} + (1-\beta)\boldsymbol{x}_{i_2}$ is in regions (ii) or (iii) then we can write $\beta\boldsymbol{x}_{i_1} + (1-\beta)\boldsymbol{x}_{i_2} = \gamma\boldsymbol{x}_{i_2} + (1-\gamma)\boldsymbol{q}$ for some $\gamma \in [0,1]$ and some $\boldsymbol{q} \in \mathcal{B}_a$.

Now, clearly $\bigcup_k \text{conv}\,(\boldsymbol{x}_k, \mathcal{B}_a) \subseteq \text{conv}\,(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M, \mathcal{B}_a)$, so we only need to show the inclusion in other direction. Let $\boldsymbol{x}$ be in $\text{conv}\,(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M, \mathcal{B}_a)$, which means that

$$\boldsymbol{x} = \beta_1^0\boldsymbol{x}_1 + \cdots \beta_M^0\boldsymbol{x}_M + (1 - \sum_k \beta_k^0)\boldsymbol{r}^0 \tag{1.7}$$

for some $\boldsymbol{r}^0 \in \mathcal{B}_a$, $\beta_k^0 \in [0,1]$ and satisfying $1 - \sum_k \beta_k^0 \in [0,1]$. Take any distinct i and j with $\beta_i^0 \neq 0$ and $\beta_j^0 \neq 0$, then we use our earlier observation to get that one of the cases below holds.

$$\frac{\beta_i^0}{\beta_i^0 + \beta_j^0}\boldsymbol{x}_1 + \frac{\beta_i^0}{\beta_i^0 + \beta_i^0}\boldsymbol{x}_2 = \begin{cases} \gamma\boldsymbol{x}_i + (1-\gamma)\boldsymbol{q} & \text{for some } \gamma \in [0,1] \text{ and some } \boldsymbol{q} \in \mathcal{B}_a, \text{ or} \\ \gamma\boldsymbol{x}_j + (1-\gamma)\boldsymbol{q} & \text{for some } \gamma \in [0,1] \text{ and some } \boldsymbol{q} \in \mathcal{B}_a. \end{cases}$$

Regardless which of these cases holds, we can substitute it back in Equation 1.7 and get a new expression

$$\boldsymbol{x} = \beta_1^1\boldsymbol{x}_1 + \cdots \beta_M^1\boldsymbol{x}_M + (1 - \sum_k \beta_k^1)\boldsymbol{r}^1,$$

where $\beta_i^1 = 0$ or $\beta_j^1 = 0$ and we still have $\beta_k \in [0,1]$ for any k. Also, we still have $(1 - \sum_k \beta_k^1) \in$

$[0,1]$ and we have $\boldsymbol{r}^1 = \frac{(\beta_i^0+\beta_j^0)(1-\gamma)\boldsymbol{q}+(1-\sum_k \beta_k^0)\boldsymbol{r}^0}{(1-\sum_i \beta_k^1)}$. We check that

$$(\beta_i^0+\beta_j^0)(1-\gamma)+(1-\sum_k \beta_k^0) = 1-\sum_{k\notin\{i,j\}} \beta_k^0 - \gamma(\beta_i^0+\beta_j^0) = 1-\sum_k \beta_k^1,$$

which means that $\boldsymbol{r}^1$ is a convex combination of $\boldsymbol{q}$ and $\boldsymbol{r}^0$, and since $\boldsymbol{q}, \boldsymbol{r}^0 \in \mathcal{B}_a$ this means that $\boldsymbol{r}^1$ is also in $\mathcal{B}_a$.

Now, further observe that the argument above has the following extra property: $\beta_k^0 = 0$ for some $k \notin \{i,j\}$, we also have $\beta_k^1 = 0$. Therefore, if we use the argument above iteratively to obtain values $(\{\beta_k^2\}, \boldsymbol{r}^2)$, $(\{\beta_k^3\}, \boldsymbol{r}^3)$ and so on, at every iteration the number of non-zero β coefficients decreases. We can keep iterating as long as there is a pair $\beta_{i'}^\ell$ and $\beta_{j'}^\ell$ both of which are nonzero, and we will terminate in M iterations or less. Thus, as we terminate we have

$$\boldsymbol{x} = \beta_{i_0}^M \boldsymbol{x}_{i_0} + \left(1-\beta_{i_0}^M\right)\boldsymbol{r}^M$$

with $\beta_{i_0}^M \in [0,1]$ and $\boldsymbol{r}^M \in \mathcal{B}_a$. This means that $\boldsymbol{x} \in \text{conv}\left(\boldsymbol{x}_{i_0}, \mathcal{B}_a\right) \subseteq \bigcup_k \text{conv}\left(\boldsymbol{x}_k, \mathcal{B}_a\right)$ finishing the proof. □

1.8.4 Proofs of main hardness theorems (theorems 5 and 6).

Proof of Theorem 5. Let δ, Δ, α and M be real-valued parameters to be chosen later. By Lemma 5 we have $\Pr_{\boldsymbol{x}\in_R\mathcal{N}(0,I_{d\times d})}\left[|\boldsymbol{x}^2| \notin [a,b]\right] \leq \alpha$, where we denote $b = \sqrt{d+2\sqrt{d\ln\left(\frac{2}{\alpha}\right)}+2\ln\left(\frac{2}{\alpha}\right)}$ and $a = \sqrt{d-2\sqrt{d\ln\left(\frac{2}{\alpha}\right)}}$.

We want to set our parameters in such a way that there is a distribution D' over $\mathbb{R}^d$ and a function $g : \mathbb{R}^d \to \{\pm 1\}$ with the following properties:

1. D' is uniform over M distinct elements $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ of $\mathbb{R}^d$.
2. A sample $\boldsymbol{x}$ from D' with probability at least $1-2\alpha$ has $|\boldsymbol{x}| \in [a,b]$.
3. Suppose $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belong to the support of D' and both $|\boldsymbol{x}_1|$ and $|\boldsymbol{x}_2|$ are in $[a,b]$. Then $|\boldsymbol{x}_1-\boldsymbol{x}_2| > 2\sqrt{b^2-a^2}$ and the line segment connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ intersects $\mathcal{B}_a$.
4. Given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each $\boldsymbol{x}_j$ i.i.d. from D', the tester $\mathcal{T}$ accepts with probability at least $1-\delta$.
5. Given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each $\boldsymbol{x}_j$ i.i.d. from D', the learner $\mathcal{A}$ outputs a predictor $\widehat{f}$ for which

$$\mathbb{E}_{\{\boldsymbol{x}_i\},\text{ randomness of }\mathcal{A}}\left[\left|\Pr_{\boldsymbol{x}\in_R D'}\left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x})\right] - \frac{1}{2}\right|\right] \leq 12\alpha + 6\frac{N}{M} + 24\sqrt{\frac{\ln M}{M}}.$$

Let D' be uniform over a multiset $S := \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ of elements drawn i.i.d. uniformly from $\mathcal{N}(0, I_{d\times d})$. Let g be a random function over $\mathbb{R}^d$ picked as follows:

- If $|\boldsymbol{x}| > b$,then $g(\boldsymbol{x}) = 0$.
- If $|\boldsymbol{x}| < a$, then $g(\boldsymbol{x}) = 1$.
- If $|\boldsymbol{x}| \in [a, b]$, then $g(\boldsymbol{x})$ is chosen randomly in $\{\pm 1\}$ subject to the following conditions.
  - For every $\boldsymbol{x}$, we have $\Pr[g(\boldsymbol{x}) = 0] = \Pr[g(\boldsymbol{x}) = 1] = \frac{1}{2}$.
  - For any collection of $\{\boldsymbol{x}_i\}$, such that any two distinct $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are further away[15] from each other than $\sqrt{b^2 - a^2}$, then $\{g(\boldsymbol{x}_i)\}$ is a collection of i.i.d. random variables uniform on $\{\pm 1\}$.

One way to give an explicit construction of random function g satisfying conditions above is to break the region $\{\boldsymbol{x} \in \mathbb{R}^s : |\boldsymbol{x}| \in [a, b]\}$ into finitely many disjoint parts of diameter at most $\sqrt{b^2 - a^2}$ and have g be i.i.d. uniformly random in $\{\pm 1\}$ on each of these parts. Then, we have

1. Condition 1 is satisfied with probability 1, because $\mathcal{N}(0, I_{d\times d})$ has continuous density.
2. By Lemma 5, for $\boldsymbol{x}$ drawn from $\mathcal{N}(0, I_{d\times d})$, the probability that $|\boldsymbol{x}| \notin [a, b]$ is at most α. Then, another application of the standard Hoeffding bound shows that out of $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$, the fraction with norm outside of $[a, b]$ is at most 2α with probability at most $e^{-2\alpha^2 M}$. In other words, Condition 2 is satisfied with probability at least $1 - e^{-2\alpha^2 M}$.
3. Claim 1 tells us that distinct i and j the probability that $|\boldsymbol{z}_i|, |\boldsymbol{z}_j| \in [a, b]$ and $|\boldsymbol{z}_i - \boldsymbol{z}_j| \leq 2\sqrt{b^2 - a^2}$ is at most $8^d \left(\frac{b^2 - a^2}{d}\right)^{d/2}$. Claim 2 then tells us that if $|\boldsymbol{z}_i - \boldsymbol{z}_j| > 2\sqrt{b^2 - a^2}$, then the line segment connecting $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ intersects $\mathcal{B}_a$. Taking a union bound over all such distinct pairs $(\boldsymbol{z}_i, \boldsymbol{z}_j)$, the probability of the Condition 3 being violated is at most $1 - 8^d \left(\frac{4\sqrt{d \ln\left(\frac{2}{\alpha}\right)} + 2\ln\left(\frac{2}{\alpha}\right)}{d}\right)^{d/2} M^2$.
4. Via Lemma 3 we see that with probability at least $1 - \Delta$ we have that given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each $\boldsymbol{x}_j$ i.i.d. from D', the tester $\mathcal{T}$ accepts with probability at least $1 - \delta_2 - \frac{N^2}{M} - \frac{N}{\sqrt{\Delta M}}$. So, to satisfy Condition 4, we need that $\delta - \delta_2 > \frac{N^2}{M}$ and $\Delta = \frac{N^2}{M} \frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2}$.
5. For any distinct $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ satisfying $|\boldsymbol{z}_i|, |\boldsymbol{z}_j| \in [a, b]$, Condition 3 tells us that $|\boldsymbol{z}_i - \boldsymbol{z}_j| > 2\sqrt{b^2 - a^2}$. The way random function g was constructed then implies that the random variables $\{g(\boldsymbol{z}_i),\ i :\ |\boldsymbol{z}_i| \in [a, b]\}$ is a collection of i.i.d. random variables uniform in $\{\pm 1\}$. We

[15]The exact value of $\sqrt{b^2 - a^2}$ here does not matter. We could have taken it to be anything smaller than $2\sqrt{b^2 - a^2}$.

can therefore use Lemma 4 as long as $2\alpha < \frac{1}{4}$ and $N \leq \frac{M}{4}$ and

$$\mathbb{E}_{g,\,\{\boldsymbol{x}_i\},\,\text{randomness of }\mathcal{A}}\left[\left|\Pr_{\boldsymbol{x}\sim D'}\left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x})\right] - \frac{1}{2}\right|\right] \leq \frac{3}{2}\left(2\alpha + \frac{N}{M}\right) + 5\sqrt{\frac{\ln M}{M}}.$$

Therefore, with probability at least $1 - \frac{1}{4}$ over the choice of g we have

$$\mathbb{E}_{\{\boldsymbol{x}_i\},\,\text{randomness of }\mathcal{A}}\left[\left|\Pr_{\boldsymbol{x}\sim D'}\left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x})\right] - \frac{1}{2}\right|\right] \leq 12\alpha + 6\frac{N}{M} + 20\sqrt{\frac{\ln M}{M}}.$$

Overall, the probability that all five of the conditions hold is non-zero as long as $\delta - \delta_2 > \frac{N^2}{M}$ and

$$e^{-2\alpha^2 M} + 8^d\left(\frac{4\sqrt{d\ln\left(\frac{2}{\alpha}\right)} + 2\ln\left(\frac{2}{\alpha}\right)}{d}\right)^{d/2} M^2 + \frac{N^2}{M}\frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2} + \frac{1}{4} < 1. \tag{1.8}$$

From now on we fix g and D' assuming the five conditions above hold (we will check that the Equation 1.8 indeed holds when we pick our parameters). We claim that there is a function $f_0 : \mathbb{R}^d \to \{\pm 1\}$ such that (i) $\{\boldsymbol{x} :\ f_0(\boldsymbol{x}) = 1\}$ is a convex set (ii) $\Pr_{\boldsymbol{x}\sim D'}\left[f_0(\boldsymbol{x}) = g(\boldsymbol{x})\right] = 1$ (even though the function g itself is very likely not indicator of a convex body). Recall that D' was uniform from $S := \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ so we define f_0 to be 1 on $\text{conv}\left(\mathcal{B}_a, \{\boldsymbol{z}_i : g(\boldsymbol{z}_i) = 1\}\right)$ and 0 otherwise. Property (i) is immediate from the definition of f_0. To show property (ii), recall that D' is supported on $\{\boldsymbol{z}_i\}$, so we need to show that $f_0(\boldsymbol{z}_i) = g(\boldsymbol{z}_i)$ for every i.

- If $g(\boldsymbol{z}_i) = 1$, from definition of f_0 it is immediate that $f_0(\boldsymbol{z}_i) = g(\boldsymbol{z}_i)$.
- If $g(\boldsymbol{z}_i) = 0$, we argue as follows. By Claim 3 we know that for any $j \neq i$ we have $\text{conv}\left(\boldsymbol{z}_i, \mathcal{B}_a\right) \cap \text{conv}\left(z_j, \mathcal{B}_a\right) = \mathcal{B}_a$ which in particular implies $\boldsymbol{z}_i \notin \text{conv}\left(\boldsymbol{z}_j, \mathcal{B}_a\right)$. So, $\boldsymbol{z}_i$ is not in $\bigcup_{j:\, g(\boldsymbol{z}_j)=1} \text{conv}\left(\boldsymbol{z}_j, \mathcal{B}_a\right)$, but $\bigcup_{j:\, g(\boldsymbol{z}_j)=1} \text{conv}\left(\boldsymbol{z}_j, \mathcal{B}_a\right) = \text{conv}\left(\{\boldsymbol{z}_j : g(\boldsymbol{z}_j) = 1\}, \mathcal{B}_a\right)$ by Claim 4, so $\boldsymbol{z}_i \notin \text{conv}\left(\{\boldsymbol{z}_j : g(\boldsymbol{z}_j) = 1\}, \mathcal{B}_a\right)$ and therefore $f_0(\boldsymbol{z}_i) = 0$ as required.

Finally, we get to picking the parameters. Recall that $N = 2^{0.01n}$. We take $M = 2^{0.1n}$ and $\delta = \delta_2 + \frac{N^2}{M} + 100\frac{N}{\sqrt{M}} = \delta_2 + 100\frac{N}{2^{0.05n}} + \frac{N^2}{2^{0.1n}}$, which allows us to conclude that the tester $\mathcal{T}$, given samples $(\boldsymbol{x}, g(\boldsymbol{x}))$ with $\boldsymbol{x} \sim D'$, accepts with probability at least $1 - \delta_2 - 100\frac{N}{2^{0.05n}} - \frac{N^2}{2^{0.1n}} = 1 - \delta_2 - \frac{1}{2^{\Omega(d)}}$. We proceed to making sure Equation 1.8 is satisfied:

- We see that $\frac{N^2}{M}\frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2} = \frac{N^2}{M}\frac{1}{10000\frac{N^2}{M}} = \frac{1}{10000}$.
- By taking $\alpha = \frac{1}{2}e^{-\frac{d}{160000}}$, we make sure that now $e^{-2\alpha^2 M} + 8^d\left(\frac{4\sqrt{d\ln\left(\frac{2}{\alpha}\right)} + 2\ln\left(\frac{2}{\alpha}\right)}{d}\right)^{d/2} M^2 = 2^{-\Omega(d)}$, so taking d sufficiently large we can make this expression as small as we want.

Thus, Equation 1.8 indeed holds for sufficiently large d for our choice of the parameters. We see that our choice of parameters also satisfies the required condition $\delta - \delta_2 > \frac{N^2}{M}$. Condition 5 tells that the expected advantage of the predictor $\widehat{f}$ is at most

$$12\alpha + 6\frac{N}{M} + 20\sqrt{\frac{\ln M}{M}} = 6e^{-\frac{d}{160000}} + 6\frac{2^{0.01n}}{2^{0.1n}} + \frac{20\sqrt{0.1n}}{2^{0.005n}} = 2^{-\Omega(d)}.$$

□

Now, let's prove our theorem about hardness of testable agnostic learning of monotone functions.

Proof of Theorem 6. Let δ, Δ, α and M be real-valued parameters to be chosen later. Observe that we have $\Pr_{\boldsymbol{x}\in\{0,1\}^d}\left[|\boldsymbol{x}| \notin \left[\frac{d}{2} - \sqrt{\frac{d}{2}\ln\frac{2}{\alpha}}\right]\right] \leq \alpha$, and denote $h_\alpha := \sqrt{\frac{d}{2}\ln\frac{2}{\alpha}}$. We want to set our parameters in such a way that there is a distribution D' over $\{0,1\}^d$ and a function $g : \{0,1\}^d \to \{\pm 1\}$ with the following properties:

1. D' is uniform over M distinct elements $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ of $\{0,1\}^d$.
2. A sample $\boldsymbol{x}$ from D' with probability at least $1-2\alpha$ has hamming weight in $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$.
3. Suppose $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belong to the support of D' and both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have hamming weight in $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$. Then $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are incomparable (i.e. neither one dominates the other one bit-wise).
4. Given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each $\boldsymbol{x}_j$ i.i.d. from D', the tester $\mathcal{T}$ accepts with probability at least $1 - \delta$.
5. Given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each $\boldsymbol{x}_j$ i.i.d. from D', the learner $\mathcal{A}$ outputs a predictor $\widehat{f}$ for which

$$\mathbb{E}_{\{\boldsymbol{x}_i\},\text{ randomness of }\mathcal{A}}\left[\left|\Pr_{\boldsymbol{x}\in_R D'}\left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x})\right] - \frac{1}{2}\right|\right] \leq 12\alpha + 6\frac{N}{M} + 20\sqrt{\frac{\ln M}{M}}.$$

We use the probabilistic method to show the existence of such D' and g. Let D' be uniform over a multiset $S := \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ of elements drawn i.i.d. uniformly from $\{0,1\}^d$. Let g be a random function over $\{0,1\}^d$ picked as

$$g(\boldsymbol{x}) = \begin{cases} 1 & \text{if } |\boldsymbol{x}| > d/2 + h_\alpha, \\ -1 & |\boldsymbol{x}| < d/2 - h_\alpha, \\ \text{i.i.d. uniformly random in } \{\pm 1\} & \text{otherwise.} \end{cases}$$

Then, we have

1. Condition 1 is satisfied with probability at least $1 - \frac{M^2}{2^d}$ by a standard birthday-paradox argument.

2. By the standard Hoeffding bound, a uniform sample from $\{0,1\}^d$ falls outside of $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$ with probability at most $2e^{-\frac{2h_\alpha^2}{d}} = \alpha$. Then, another application of the standard Hoeffding bound shows that out of $\{z_1, \cdots, z_M\}$, the fraction with Hamming weight outside of $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$ is at most 2α with probability at most $e^{-2\alpha^2 M}$. In other words, Condition 2 is satisfied with probability at least $1 - e^{-2\alpha^2 M}$.

3. For distinct i_1 and i_2, we bound the probability probability of the event that (i) $\boldsymbol{z}_{i_1}$ and $\boldsymbol{z}_{i_2}$ have Hamming weight in $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$ and (ii) $\boldsymbol{z}_{i_1}$ dominates $\boldsymbol{z}_{i_2}$ bit-wise. Suppose $\boldsymbol{z}_{i_1}$ indeed has Hamming weight in $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$, then there are only at most d^{2h_α} possible candidates for $\boldsymbol{z}_{i_2}$ that will make the event to take place. Thus, the probability of this event is at most $\frac{d^{2h_\alpha}}{2^d} = \frac{d^{\sqrt{2n \ln \frac{2}{\alpha}}}}{2^d}$. Taking a union bound over all distinct pairs $(\boldsymbol{z}_{i_1}, \boldsymbol{z}_{i_2})$, the probability of the Condition 3 being violated is at most $1 - \frac{d^{\sqrt{2n \ln \frac{2}{\alpha}}}}{2^d} M^2$.

4. Via Lemma 3 we see that with probability at least $1 - \Delta$ we have that given N samples of the form $(\boldsymbol{x}_j, g(\boldsymbol{x}_j))$ with each x_j i.i.d. from D', the tester $\mathcal{T}$ accepts with probability at least $1 - \delta_2 - \frac{N^2}{M} - \frac{N}{\sqrt{\Delta M}}$. So, to satisfy Condition 4, we need that $\delta - \delta_2 > \frac{N^2}{M}$ and $\Delta = \frac{N^2}{M} \frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2}$.

5. Via Lemma 4 we have

$$\mathbb{E}_{g,\, \{\boldsymbol{x}_i\},\, \text{randomness of } \mathcal{A}} \left[\left| \Pr_{\boldsymbol{x} \sim D'} \left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x}) \right] - \frac{1}{2} \right| \right] \leq \frac{3}{2} \left(2\alpha + \frac{N}{M} \right) + 6\sqrt{\frac{\ln M}{M}}.$$

   Therefore, with probability at least $1 - \frac{1}{4}$ over the choice of g we have

$$\mathbb{E}_{\{\boldsymbol{x}_i\},\, \text{randomness of } \mathcal{A}} \left[\left| \Pr_{\boldsymbol{x} \sim D'} \left[\widehat{f}(\boldsymbol{x}) \neq g(\boldsymbol{x}) \right] - \frac{1}{2} \right| \right] \leq 12\alpha + 6\frac{N}{M} + 24\sqrt{\frac{\ln M}{M}}.$$

Overall, the probability that all five of the conditions hold is non-zero as long as $\delta - \delta_2 > \frac{N^2}{M}$ and

$$\frac{M^2}{2^d} + e^{-2\alpha^2 M} + \frac{d^{\sqrt{2n \ln \frac{2}{\alpha}}}}{2^d} M^2 + \frac{N^2}{M} \frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2} + \frac{1}{4} < 1. \tag{1.9}$$

From now on, we assume that the five conditions above hold (we will check that the Equation 1.9 indeed holds when we pick our parameters). We claim that there is a monotone $f_0 : \{0,1\}^d \to \{\pm 1\}$ for which $\Pr_{\boldsymbol{x} \sim D'} [f_0(\boldsymbol{x}) = g(\boldsymbol{x})] = 1$ (even though the function g itself is very likely not

monotone). Recall that D' was uniform from $S := \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_M\}$ so we write

$$f_0(\boldsymbol{x}) = \begin{cases} 1 & \text{if } |\boldsymbol{x}| > d/2 + h_\alpha, \\ -1 & |\boldsymbol{x}| < d/2 - h_\alpha, \\ g(\boldsymbol{x}) & \text{if } |\boldsymbol{x}| \in [d/2 - h_\epsilon, d/2 + h_\epsilon] \text{ and } \boldsymbol{x} \text{ is in the support of } D', \\ -1 & \text{if } |\boldsymbol{x}| \in [d/2 - h_\epsilon, d/2 + h_\epsilon] \text{ and } \boldsymbol{x} \preceq \boldsymbol{y} \text{ for some } \boldsymbol{y} \text{ in supp}(D') \text{ s.t. } g(\boldsymbol{y}) = -1, \\ 1 & \text{if } |\boldsymbol{x}| \in [d/2 - h_\epsilon, d/2 + h_\epsilon] \text{ and } \boldsymbol{x} \succeq \boldsymbol{y} \text{ for some } \boldsymbol{y} \text{ in supp}(D') \text{ s.t. } g(\boldsymbol{y}) = +1, \\ -1 & \text{otherwise.} \end{cases}$$

The definition above is not self-contradictory, because Condition 3 says if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belong to the support of D' and both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have hamming weight in $\left[\frac{d}{2} - h_\alpha, \frac{d}{2} + h_\alpha\right]$, then $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are incomparable. We see that $f_0(\boldsymbol{x})$ is indeed monotone and agrees with g on the support of D.

Finally, we get to picking the parameters. Recall that $N = 2^{0.01n}$. We take $M = 2^{0.1n}$ and $\delta = \delta_2 + \frac{N^2}{M} + 100\frac{N}{\sqrt{M}} = \delta_2 + 100\frac{N}{2^{0.05n}} + \frac{N^2}{2^{0.1n}}$, which allows us to conclude that the tester $\mathcal{T}$, given samples $(\boldsymbol{x}, g(\boldsymbol{x}))$ with $\boldsymbol{x} \sim D'$, accepts with probability at least $1 - \delta_2 - 100\frac{N}{2^{0.05n}} - \frac{N^2}{2^{0.1n}} = 1 - \delta_2 - \frac{1}{2^{\Omega(d)}}$. We proceed to making sure Equation 1.9 is satisfied:

- We see that $\frac{N^2}{M}\frac{1}{\left(\delta - \delta_2 - \frac{N^2}{M}\right)^2} = \frac{N^2}{M}\frac{1}{10000\frac{N^2}{M}} = \frac{1}{10000}$.
- Taking $\alpha = \frac{1}{2}e^{-0.1\frac{d}{\log^2 d}}$, we see that now $\frac{M^2}{2^d} + e^{-2\alpha^2 M} + \frac{d\sqrt{2n\ln\frac{2}{\alpha}}}{2^d}M^2 = 2^{-\Omega(d)}$, so taking d sufficiently large we can make this expression as small as we want.

Thus, Equation 1.9 holds for sufficiently large d for our choice of the parameters. We see that our choice of parameters also satisfies the required condition $\delta - \delta_2 > \frac{N^2}{M}$. Condition 5 tells that the expected advantage of the predictor $\widehat{f}$ is at most

$$12\alpha + 6\frac{N}{M} + 20\sqrt{\frac{\ln M}{M}} = 6e^{-0.1\frac{d}{\log^2 d}} + 6\frac{2^{0.01n}}{2^{0.1n}} + \frac{20\sqrt{0.1n}}{2^{0.005n}} = 2^{-\Omega\left(\frac{d}{\log^2 d}\right)}.$$

□

1.9 Miscellaneous proofs.

1.9.1 Improvement of error probabilities for a tester-learner pair via repetition.

If the constants $1/4$ and $3/4$ in the Definition 3 are replaced by some other constants $1 - \delta_2$ and $1 - \delta_3$ with $\delta_3 \in (\delta_2, 1)$, then we say that $\mathcal{T}$ is a (δ_2, δ_3)-*tester for the distributional assumption* of $\mathcal{A}$. The following proposition tells us that that taking $\delta_2 = 1/4$ and $\delta_3 = 3/4$ is without loss of generality.

Proposition 18. *Let $\delta_1, \delta_2, \epsilon \in (0,1)$, $\delta_3 \in (\delta_2, 1)$ and let $\mathcal{A}$ be an agnostic (ϵ, δ_1)-learner for function class $\mathcal{F}$ relative to the distribution D, and $\mathcal{T}$ be a (δ_2, δ_3)-tester for the distributional assumption of $\mathcal{A}$. Then, for every integer $r \geq 1$ there is a $\left(2\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right), 1-3\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right)\right)$-tester $\mathcal{T}'$ for the distributional assumption of $\mathcal{A}$ that consumes only $O(r)$ times as much samples and run-time as $\mathcal{T}$.*

Proof. The tester $\mathcal{T}'$ is constructed by (i) repeating $\mathcal{T}$ r times (ii) if the fraction of "Yes" answers is at least $1-\frac{\delta_2+\delta_3}{2}$, then output "Yes", otherwise output "No". By Hoeffding's bound with probability at least $1-2\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right)$, the fraction of "Yes" answers observed is within $\frac{\delta_3-\delta_2}{3}$ of the true probability that $\mathcal{T}$ outputs "Yes". So,

- Recall that, given access to samples from D_{pairs}, the algorithm $\mathcal{T}$ outputs "Yes" with probability at least $1-\delta_2$. Therefore, given access to samples from D_{pairs}, the algorithm $\mathcal{T}'$ outputs "Yes" with probability at least $1-2\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right)$.
- Suppose, given access to samples from D_{pairs}, the algorithm $\mathcal{T}$ outputs "Yes" with probability p. Then, if $p < 1-\delta_3$, the algorithm $\mathcal{T}'$ can output "Yes" with probability only at most $2\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right)$. Therefore, if the algorithm $\mathcal{T}'$ outputs "Yes" with probability at least $3\exp\left(-\frac{2(\delta_3-\delta_2)^2 r}{9}\right)$, it has to be the case that the algorithm $\mathcal{T}$ outputs "Yes" with probability at least $1-\delta_3$. The Soundness condition then tells us that $\mathcal{A}$ will then satisfy the required bound on the generalization error when run on samples from D_{pairs}.

□

1.9.2 Proof of Observation 1.

Since for $y \in [-1,1]$ both $f(wy)$ and $T_k(y)$ are also in $[-1,1]$, we have that[16] all a_k are in $[-4,4]$. We also see that that the largest coefficient among all the monomials of $T_k(y)$ is at most 3^k (this follows by induction via the recursive relation $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$). Since $w \geq 1$, the largest coefficient among all the monomials of $T_k\left(\frac{y}{w}\right)$ is also at most 3^k. Thus, the largest coefficient of $f_s(x) := \sum_{k=0}^{s} a_k T_k(\frac{x}{w})$ can only be at most $O(s3^s)$.

[16]Proof: we have $|a_k| = \left|\frac{1+\mathbb{1}_{k>0}}{\pi}\int_{-1}^{1}\frac{f(wy)T_k(y)}{\sqrt{1-y^2}}\,dy\right| \leq \frac{2}{\pi}\int_{-1}^{1}\frac{1}{\sqrt{1-y^2}}\,dy = 4$, where the integral in the end is evaluated via a standard substitution of $y = \cos(\alpha)$.

1.9.3 Proof of Proposition 1.

We have

$$\mathbb{E}_{x\in_R D}\left[|f(x)-f_d(x)|\,\mathbb{1}_{|x|>w}\right]\overbrace{\leq \mathbb{E}_{x\in_R D}\left[(1+|f_d(x)|)\,\mathbb{1}_{|x|>w}\right]}^{\text{Since } |f(x)|\leq 1.} \leq$$

$$\overbrace{\leq \mathbb{E}_{x\in_R D}\left[\mathbb{1}_{|x|>w}\right]+O\left(d3^d\right)\sum_{k=0}^{d}\mathbb{E}_{x\in_R D}\left[|x|^k\,\mathbb{1}_{|x|>w}\right]}^{\text{Breaking } f_s \text{ into monomials, then using triangle inequality and Observation 1}}$$

$$\underbrace{\leq O\left(4^d\max_{0\leq k\leq d}\mathbb{E}_{x\in_R D}\left[|x|^k\,\mathbb{1}_{|x|>w}\right]\right)=O\left(4^d\mathbb{E}_{x\in_R D}\left[|x|^d\,\mathbb{1}_{|x|>w}\right]\right)}_{\text{Since } w\geq 1\text{, when } |x|>w \text{ the value of } |x|^k \text{ grows with } k.}$$

1.9.4 Proof of Proposition 2.

Applying Markov's inequality to $|x|^{d_0}$, we have

$$\Pr\left[|x|\geq\tau\right]\leq\left(\frac{\beta}{\tau}\right)^{d_0}.$$

The above covers the case when $k=0$. When $k>0$ we proceed by using the inequality above as follows,

$$\mathbb{E}_{x\in_R D}\left[|x|^k\,\mathbb{1}_{|x|>w}\right]=\mathbb{E}_{x\in_R D}\left[w^k\mathbb{1}_{|x|>w}+\int_{\tau=w}^{\infty}k\tau^{k-1}\mathbb{1}_{|x|>\tau}\,dt\right]=$$

$$\underbrace{=w^k\Pr_{x\in_R D}\left[|x|>w\right]+\int_{w}^{\infty}k\tau^{k-1}\Pr_{x\in_R D}\left[|x|>\tau\right]\,d\tau}_{\text{Via linearity of expectation and Tonelli's theorem.}}\leq$$

$$w^k\left(\frac{\beta}{w}\right)^{d_0}+\int_{w}^{\infty}k\tau^{k-1}\left(\frac{\beta}{\tau}\right)^{d_0}d\tau=w^k\left(\frac{\beta}{w}\right)^{d_0}+\frac{k}{d_0-k}w^k\left(\frac{\beta}{w}\right)^{d_0}=$$

$$=w^k\left(\frac{\beta}{w}\right)^{d_0}\frac{d_0}{d_0-k}\leq 2w^k\left(\frac{\beta}{w}\right)^{d_0}.$$

1.9.5 Proof of Observation 2.

Without loss of generality, assume $\Delta=\frac{1}{\epsilon^4}\ln^4\left(\frac{1}{\epsilon}\right)$.

We have

$$\left|\mathbb{E}_{\boldsymbol{x}\sim D}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^d\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[(\boldsymbol{v}\cdot\boldsymbol{x})^d\right]\right| = \left|\sum_{\substack{\alpha_1,\cdots\alpha_n\in\mathbb{Z}_{\geq 0}\\ \alpha_1+\cdots+\alpha_n=d}}\left(\mathbb{E}_{\boldsymbol{x}\sim D}\left[\prod_{i=1}^n(v_ix_i)^{\alpha_i}\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{N}(0,I_{n\times n})}\left[\prod_{i=1}^n(v_ix_i)^{\alpha_i}\right]\right)\right| \leq \underbrace{\sum_{\substack{\alpha_1,\cdots\alpha_n\in\mathbb{Z}_{\geq 0}\\ \alpha_1+\cdots+\alpha_n=d}}\left(\left|\prod_{i=1}^n v_i^{\alpha_i}\right|\frac{1}{n^\Delta}\right)}_{\text{Using } |v_i|\leq 1 \text{ and bounding the number of } \{\alpha_i\}} \leq n^d\frac{1}{n^\Delta}.$$

1.9.6 Tester-learner pair for decision lists

First, recall the definition of a decision list (a more general definition is given in [Riv87]):

Definition 5. *For some ordering of the variables $x_{\pi(1)},\ldots,x_{\pi(d)}$, values $v_1,\ldots,v_n\in\{\pm1\}$ and bits $b_1,\ldots,b_n\in\{\pm1\}$, a **decision list** does the following: For $i=1$ to d, if $x_{\pi(i)}=b_{\pi(i)}$ output $v_{\pi(i)}$, else continue. If the decision list reaches the end of execution without outputting anything, it outputs* 0.

Now, we will present the tester-learner pair for decision lists. The tester will check that the distribution on examples is close to k-wise independent. The insight behind the learning algorithm is that any decision list is well-approximated by a short decision list if the distribution on examples is close to k-wise independent.

Tester-learner pair for learning decision lists over $\{0,1\}^d$:

- Define $k:=\log\frac{1}{\epsilon}$.
- **Learning algorithm $\mathcal{A}_{\mathbf{DL}}$.** Given access to i.i.d. labeled samples $(\boldsymbol{x},y)\in\{\pm1\}^d\times\{\pm1\}$ from an unknown distribution:
  - Take $\frac{100}{\epsilon^2}k^3\log d$ samples $\{(\boldsymbol{x}_i,y_i)\}$.
  - Enumerate over all functions f that can be represented as decision lists on any size-k subset S of $\{1,\cdots d\}$: Compute the fraction of example label pairs on which f gives the wrong answer. Denote it as $\widehat{\text{err}}(f)$.
  - Among the functions just considered, output the function f_0 that fits best. In other words, the function for which $\widehat{\text{err}}(f)$ was smallest.
- **Testing algorithm $\mathcal{T}_{\mathbf{DL}}$.** Given access to i.i.d. labeled examples $\boldsymbol{x}\in\{\pm1\}^d$ from an unknown distribution:

1. Use a tester from literature (see [OZ18, AAK+07, AGM03]) for testing k-wise independent distributions against distributions that are ϵ-far from k-wise independent.
2. Output the same response as the one given by the k-wise independence tester.

Theorem 7 (**Tester-learner pair for learning decision lists under uniform distribution on** $\{\pm 1\}^d$)**.** *Assume d and $\frac{1}{\epsilon}$ are larger than some sufficiently large absolute constants. Then, the algorithm $\mathcal{A}_{DL}$ is an agnostic $(O(\epsilon), 0.1)$-learner for the function class of decision lists (see Definition 5) over $\{\pm 1\}^d$ under the uniform distribution and the algorithm $\mathcal{T}_{DL}$ is an assumption tester for $\mathcal{A}_{DL}$. The algorithms $\mathcal{A}_{DL}$ and $\mathcal{T}_{DL}$ both require only $d^{O\left(\log \frac{1}{\epsilon}\right)}$ samples and run-time. Additionally, the tester $\mathcal{T}_{DL}$ is label-oblivious.*

The testers from the literature for k-wise independence take $d^{O(k)}/\eta^2$ samples and run-time to distinguish a k-wise independent distribution and a distribution that is η-far from k-wise independent (see [OZ18, AAK+07, AGM03]). Thus, the run-time of tester $\mathcal{T}_{\mathrm{DL}}$ is $d^{O(\log(1/\epsilon))}$. The same run-time bound of $d^{O(\log(1/\epsilon))}$ holds for $\mathcal{A}_{\mathrm{DL}}$ for the following reason. There are only at most d^k of size-k subsets of $\{1, \cdots, d\}$ and there are at most $2^k \cdot k^k$ decision lists on each size-k set. Substituting $k = \log(1/\epsilon)$ gives a bound of $d^{O(\log(1/\epsilon))}$ on the number of functions f considered by the algorithm and hence on the run-time.

The only thing remaining to prove is that the algorithm $\mathcal{A}_{\mathrm{DL}}$ is indeed a $(O(\epsilon), 0.1)$-agnostic learning algorithm for the class of decision lists on $\{\pm 1\}^d$ with respect to distributions D that are ϵ-close to k-wise independent.

Let D_{pairs} be the distribution from which we are getting example-label pairs. For any function $g : \{\pm 1\}^d \to \{\pm 1\}$ we let *error of* g denote the flowing:

$$\mathrm{err}(g) := \Pr_{(\boldsymbol{x},y)\sim D_{\text{pairs}}}[g(\boldsymbol{x}) \neq y]$$

Let opt be the smallest error among all decision lists. We want to show that if the distribution of examples is ϵ-close to k-wise independent, then the function f_0 that $\mathcal{A}_{\mathrm{DL}}$ outputs has $\mathrm{err}(f_0)$ that is at most $\mathrm{opt} + O(\epsilon)$.

First of all, by the Hoeffding bound and the union bound, we have that with probability at least 0.9 for every function considered by the algorithm $\mathcal{A}_{\mathrm{DL}}$ it is the case that

$$|\widehat{\mathrm{err}}(f) - \mathrm{err}(f)| \leq O(\epsilon).$$

Thus, the only thing left to prove is that among the functions f considered by $\mathcal{A}_{\mathrm{DL}}$ there is one for which $\mathrm{err}(f)$ is at most $\mathrm{opt} + O(\epsilon)$. That follows from the following proposition:

Proposition 19. *Let g be a decision list over $\{\pm 1\}^d$ and let $k := \log \frac{1}{\epsilon}$. Also let D be a distribution that is ϵ-close in TV distance to k-wise independent. Then, there is a decision list f on a size-k subset of $\{1, \cdots d\}$ for which*

$$\Pr_{\boldsymbol{x}\sim D}[g(x) \neq f(x)] = O(\epsilon)$$

Proof. First, we recall the definition of a decision list. For some ordering of the variables $x_{\pi(1)}, \ldots, x_{\pi(d)}$, values $v_1, \ldots, v_n \in \{\pm 1\}$ and bits $b_1, \ldots, b_n \in \{\pm 1\}$, the decision list g does the following: For $i = 1$ to d, if $x_{\pi(i)} = b_{\pi(i)}$ it outputs $v_{\pi(i)}$, else it continues.

Let the decision list g be defined on the first k variables in the ordering $x_{\pi(1)}, \ldots, x_{\pi(d)}$ and let g repeat the same comparisons and outputs as f until it reaches the $k+1$-st variable.

Recall that D is only ϵ-close in TV distance to a k-wise independent distribution. Let D' be the closest k-wise independent distribution to D. Then, D' is uniform on the first k variables in the ordering on $x_{\pi(1)}, \ldots, x_{\pi(d)}$. Therefore, the execution of f will reach past the k-th comparison only with probability at most $2^k = O(\epsilon)$. The same is true for function g and therefore we have

$$\Pr_{\boldsymbol{x} \sim D'}[g(x) \neq f(x)] = O(\epsilon).$$

But from the definition of the TV distance we have that the function $\mathbb{1}_{g(x) \neq f(x)}$ should not allow us to distinguish D and D' with advantage better than ϵ. Therefore

$$|\Pr_{\boldsymbol{x} \sim D'}[g(x) \neq f(x)] - \Pr_{\boldsymbol{x} \sim D}[g(x) \neq f(x)]| \leq \epsilon.$$

Together with the previous equation we conclude

$$\Pr_{\boldsymbol{x} \sim D}[g(x) \neq f(x)] = O(\epsilon).$$

□

Chapter 2

Tester-Learners for Halfspaces: Universal Algorithms.

2.1 Chapter Overview.

In this chapter we continue our investigation of tester-learner pairs. Chapter 1 and [GKK23] establish foundational algorithmic and statistical results for this framework and show that testable learning is in general provably harder than ordinary distribution-specific agnostic learning. One of the main algorithmic results in Chapter 1 and [GKK23] are tester-learners for the class of halfspaces over $\mathbb{R}^d$ that succeed whenever the target marginal is Gaussian (or one of a more general class of distributions), achieving error $\mathrm{opt} + \epsilon$ in time[1] and sample complexity $d^{\widetilde{O}(1/\epsilon^2)}$. This matches the running time of ordinary distribution-specific agnostic learning of halfspaces over the Gaussian using the standard approach of [KKMS08]. These testers are simple and label-oblivious, and are based on checking whether the low-degree empirical moments of the unknown marginal match those of the target D^*.

These works essentially resolve the question of designing tester-learners achieving error $\mathrm{opt}+\epsilon$ for halfspaces, matching known hardness results for (ordinary) agnostic learning [GGK20, DKZ20, DKPZ21]. Their running time, however, necessarily scales exponentially in $1/\epsilon$. A long line of research has sought to obtain more efficient algorithms at the cost of relaxing the optimality guarantee [ABL14, DKS18, DKTZ20a, DKTZ20b]. These works give polynomial-time algorithms achieving bounds of the form $\mathrm{opt} + \epsilon$ and $O(\mathrm{opt}) + \epsilon$ for the Massart and agnostic setting respectively under structured distributions (see Section 2.1.1 for more discussion). The first question we consider in this chapter is whether such guarantees can be obtained in the testable learning framework.

In this chapter we design the first tester-learners for halfspaces that run in fully polynomial time in all parameters. We match the optimality guarantees of fully polynomial-time learning algorithms under Gaussian marginals for the Massart noise model (where the labels arise from a

[1]Note that Chapter 1 ahieves a run-time and sample complexity of $d^{\widetilde{O}(1/\epsilon^4)}$, which was improved to $d^{\widetilde{O}(1/\epsilon^2)}$ in [GKK23].

halfspace but are flipped by an adversary with probability at most η) as well as for the agnostic model (where the labels can be completely arbitrary). In fact, for the Massart setting our guarantee holds with respect to any chosen target marginal D^* that is isotropic and strongly log-concave, and the same is true of the agnostic setting albeit with a slightly weaker guarantee.

We also address another shortcoming of tester-learners in Chapter 1 and [GKK23], namely that they are closely tailored to the particular target marginal D^* that is chosen. Indeed, their tests would reject many well-behaved distributions that are appreciably far from D^*. A highly natural question from both a theoretical and a practical perspective is: can we design tester-learners that accept a wide class of distributions simultaneously, without being tailored to any particular one? In this work we answer this question in the affirmative by introducing and studying *universally testable learning*. We formally define this framework as follows.

Definition 6 (Universally Testable Learning)**.** *Let $\mathcal{C}$ be a concept class mapping $\mathbb{R}^d$ to $\{\pm 1\}$. Let $\mathcal{D}$ be a family of distributions over $\mathbb{R}^d$. Let $\epsilon, \delta > 0$ be parameters, and let $\psi : [0,1] \to [0,1]$ be some function. We say $\mathcal{C}$ can be universally testably learned w.r.t. $\mathcal{D}$ up to error $\psi(\mathrm{opt}) + \epsilon$ with failure probability δ if there exists a tester-learner A meeting the following specification. For any distribution $D_{\mathcal{XY}}$ on $\mathbb{R}^d \times \{\pm 1\}$, A takes in a large sample S drawn from $D_{\mathcal{XY}}$, and either rejects S or accepts and produces a hypothesis $h : \mathbb{R}^d \to \{\pm 1\}$ such that:*

1. *Soundness: With probability at least $1 - \delta$ over the sample S the following is true: If A accepts, then the output h satisfies $\mathbb{P}_{(\mathbf{x},y)\sim D_{\mathcal{XY}}}[h(\mathbf{x}) \neq y] \leq \psi(\mathrm{opt}(\mathcal{C}, D_{\mathcal{XY}})) + \epsilon$, where $\mathrm{opt}(\mathcal{C}, D_{\mathcal{XY}}) = \inf_{f\in\mathcal{C}} \mathbb{P}_{(\mathbf{x},y)\sim D_{\mathcal{XY}}}[h(\mathbf{x}) \neq y]$.*
2. *Completeness: Whenever the marginal of $D_{\mathcal{XY}}$ lies within $\mathcal{D}$, A accepts with probability at least $1 - \delta$ over the sample S.*

In this terminology, the original definition of testable learning reduces to the special case where $\mathcal{D} = \{D^*\}$, which was introduced in Chapter 1. We stress that while the work of [GKK23] allowed D^* to be, say, any fixed strongly log-concave distribution, their tester-learners are still tailored to the particular D^* that is selected. This is because their tests rely on checking that the unknown distribution closely matches moments with D^*. By contrast, a universal tester-learner must accept *all* marginals in a family $\mathcal{D}$.

This chapter contributes the first universal tester-learner for the class of halfspaces with respect to a broad family of structured continuous distributions. This family is the set of all distributions with bounded Poincaré constant (see Definition 9) and some mild concentration and anti-concentration properties (see Definition 7). It captures all strongly log-concave distributions, and in fact, under the well-known Kannan–Lóvasz–Simonovits (KLS) conjecture (see Conjecture 10), it captures all log-concave distributions as well.

Theorem 8 (Universal and Efficient Tester-Learner for Halfspaces; formally stated as Theorem 11)**.** *Let $\mathcal{C}$ be the class of origin-centered halfspaces over $\mathbb{R}^d$. Let $\mathcal{D}$ be the class of $\Theta(1)$-nice and $\Theta(1)$-Poincaré distributions (see Definitions 7 and 9), which includes all isotropic strongly log-concave and, under KLS, all isotropic log-concave distributions. Then $\mathcal{C}$ can be universally testably learned w.r.t. $\mathcal{D}$ up to error $O(\mathrm{opt}) + \epsilon$ in $\mathrm{poly}(d, \frac{1}{\epsilon})$ time and sample complexity.*

A special and well-studied case of interest is when the label noise follows the Massart model, i.e. the label of every example is flipped by an adversary with probability at most η. In this case we are able to handle a considerably larger class $\mathcal{D}$ while also providing a stronger guarantee.

Theorem 9 (Universal Tester-Learner for Massart Halfspaces; formally stated as Theorem 11)**.** *Let $\mathcal{C}$ be the class of origin-centered halfspaces over $\mathbb{R}^d$. Let $\mathcal{D}$ be the class of $\mathrm{poly}(d)$-nice and $\mathrm{poly}(d)$-Poincaré distributions, which includes all isotropic log-concave distributions (unconditionally). Suppose the label noise follows the Massart model with noise rate at most $\eta < \frac{1}{2}$. Then $\mathcal{C}$ can be universally testably learned w.r.t. $\mathcal{D}$ up to error $\mathrm{opt} + \epsilon$ in $\mathrm{poly}(d, \frac{1}{\epsilon}, \frac{1}{1-2\eta})$ time and sample complexity.*

Technical Overview. We first describe the key reasons why prior tester-learners were tailored to a specific target D^*. All known polynomial-time algorithms for agnostically learning halfspaces up to error $O(\mathrm{opt}) + \epsilon$ require some concentration and anti-concentration properties from the input marginal distribution (encapsulated e.g. in Definition 7). While concentration is relatively straightforward to check (e.g. by checking that the moments do not grow at too fast a rate), the key challenge in designing tester-learners for halfspaces is to check anti-concentration. All prior tester-learners [RV23, GKK23, DKK+23] use the heavy machinery of moment-matching to achieve this. This approach relies on establishing structural properties of the following type: if D^* is a well-behaved distribution (e.g. a strongly log-concave distribution), and D approximately matches D^* in its low-degree moments, then D is also well-behaved (in particular, anti-concentrated). A canonical statement of such a property is the main pseudorandomness result of [GKK23] (see Theorem 5.6 therein), which establishes that approximate moment-matching fools functions of a constant number of halfspaces. Applying this property inherently requires comparing the low-degree moments of D with those of D^*. Such tests do (implicitly) succeed universally for the class of all distributions that match low-degree moments with D^* (e.g., if D^* is the uniform distribution over the hypercube, moment matching would accept all k-wise independent distributions). Definition 6, however, seeks a far broader kind of universality. Our tests are not tailored to a single target in any way, and are intended to succeed over practical classes of distributions that are commonly considered in learning theory (e.g., log-concave distributions).[2]

The tester-learner first computes a stationary point $\mathbf{w}$ of a certain smooth version of the ramp loss, a surrogate for the 0-1 loss. Let $\mathbf{w}^*$ be any solution achieving 0-1 error opt. The tester-learner now checks distributional properties of the unknown marginal D that ensure that $\mathbf{w}$ is close in angular distance to $\mathbf{w}^*$ (specifically, they ensure the contrapositive, namely that any $\mathbf{w}$ that has large gradient norm must have large angle with $\mathbf{w}^*$). By a more careful analysis of the gradient norm (see Proposition 25), we are able to reduce to showing the following weak anti-concentration property. Let $\mathbf{v}$ denote any unit vector orthogonal to $\mathbf{w}$, and let D_T denote D restricted to the band $T = \{\mathbf{x} \mid |\langle \mathbf{w}, \mathbf{x}\rangle| \le \sigma\}$ (where the width σ is carefully selected according to certain constraints).

[2]One may wonder if it is possible to test whether the low-degree moments of the input marginal D match *any* distribution in a family $\mathcal{D}$ (e.g., all strongly log-concave distributions) without directly comparing to a specific D^*. This is a reduction to testing whether a given (approximate) low-degree moment tensor lies within a large set of target low-degree moment tensors, and would indeed suffice for universally testable learning. This general problem, however, seems highly challenging to solve directly.

Then the property we need is that

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim D_T}[|\langle \mathbf{v}, \mathbf{x}\rangle| \geq \Theta(1)] \geq \Theta(1).$$

Our key observation is that the classical Paley–Zygmund inequality applied to the random variable $Z = \langle \mathbf{v}, \mathbf{x}\rangle^2$, where $\mathbf{x} \sim D_T$, already gives us the following type of anti-concentration:

$$\mathbb{P}\left[Z > \frac{\mathbb{E}[Z]}{2}\right] \geq \frac{1}{4} \cdot \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}.$$

This turns out to suffice for our purposes—provided we can show a hypercontractivity property for Z, namely that $\mathbb{E}[Z^2] \leq \Theta(1)\,\mathbb{E}[Z]^2$ (as well as that $\mathbb{E}[Z] = \Theta(1)$, which is just a second moment constraint).

Our main algorithmic idea is to use a sum-of-squares (SOS) program to check hypercontractivity of the random variable Z. To do so, we crucially leverage a result due to [KS17] stating that any D that has bounded Poincaré constant is *certifiably hypercontractive* in the SOS framework (and it turns out this extends to D_T as well). This means that we can run a certain polynomial-time semidefinite program that checks hypercontractivity of Z over the sample, and whenever D is in fact Poincaré, we are guaranteed that the test will pass with high probability (see Proposition 24). This is sufficient to ensure that the stationary point $\mathbf{w}$ we have computed is indeed close in angular distance to $\mathbf{w}^*$.

In order to finally arrive at our main results, we need to run further tests which ensure that the disagreement between our computed $\mathbf{w}$ and any (unknown) optimum $\mathbf{w}^*$ is bounded by the angle between them, i.e., $\mathbb{P}_{\mathbf{x}\sim D}[\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}\rangle) \neq \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle)] \leq O(\measuredangle(\mathbf{w}, \mathbf{w}^*))$ (see Lemma 6). This in turn guarantees that $\mathbf{w}$ has error $O(\mathrm{opt}) + \epsilon$. From a technical perspective, prior to this chapter, such tests either produced a suboptimal bound, or required estimating the operator norms of a polynomial number of random matrices formed using rejection sampling. We significantly simplify this approach by showing that it is sufficient to estimate the operator norm of a single random matrix. Finally, to obtain our improved results for the Massart setting, it turns out that the proof admits certain simplifications that guarantee final error $\mathrm{opt} + \epsilon$ while also allowing a wider range of Poincaré distributions.

Related Work. There is a large body of work on agnostic learning algorithms for halfspaces that run in fully polynomial time. We briefly mention only those that are most closely relevant to this chapter; please see [BH21] for a survey.

2.1.1 Related work

In the distribution-specific agnostic setting where the marginal is assumed to be isotropic and log-concave, [KLS09] showed an algorithm achieving error $O(\mathrm{opt}^{1/3}) + \epsilon$ for the class of origin-centered halfspaces. [ABL14] later obtained $O(\mathrm{opt}) + \epsilon$ using an approach that introduced the principle of iterative *localization*, where the learner focuses attention on a band around a candidate halfspace in order to produce an improved candidate. [Dan15] used this principle to obtain

a PTAS for agnostically learning halfspaces under the uniform distribution on the sphere, and [BZ17] extended it to more general s-concave distributions. Further works in this line include [YZ17, Zha18, ZSA20, ZL21]. [DKTZ20b] introduced the simplest approach yet, based entirely on nonconvex SGD, and showed that it achieves $O(\mathrm{opt}) + \epsilon$ for origin-centered halfspaces over a wide class of structured distributions. Other related works include [DKS18, DKTZ22].

In the Massart noise setting with noise rate bounded by η, work of [DGT19] gave the first efficient distribution-free algorithm achieving error $\eta + \epsilon$; further improvements and followups include [DKT21, DTK22]. However, the optimal error opt achievable by a halfspace may be much smaller than η, and it has been shown that there are distributions where achieving error competitive with opt as opposed to η is computationally hard [DK22, DKMR22]. As a result, the distribution-specific setting remains well-motivated for Massart noise. Early distribution-specific algorithms were given by [ABHU15, ABHZ16], but a key breakthrough was the nonconvex SGD approach introduced by [DKTZ20a], which achieved error $\mathrm{opt} + \epsilon$ for origin-centered halfspaces efficiently over a wide range of distributions. This was later generalized by [DKK+22].

Following a long line of work on distribution-specific agnostic learners for halfspaces [KLS09, ABL14, Dan15, BZ17, YZ17, Zha18, ZSA20, ZL21], the work of [DKTZ20a] introduced a particularly simple approach for the Massart setting, based solely on non-convex SGD. This chapter, which sets the template that our approach also follows, achieved the information-theoretically optimal error of $\mathrm{opt} + \epsilon$ for origin-centered Massart halfspaces over a wide range of structured distributions (and was later extended to general halfspaces by [DKK+22]). The non-convex SGD approach was then generalized by [DKTZ20b] to show an $O(\mathrm{opt}) + \epsilon$ guarantee for the fully agnostic setting.

Certifying distributional properties such as hypercontractivity is an important aspect of a large body of work on robust algorithmic statistics using the SOS framework. We will not attempt to summarize this literature here and direct the reader to [KS17, BK21] for overviews of related work, as well as to [FKP+19] for a textbook treatment. The notion of certifiable anti-concentration has also been studied (see e.g. [KKK19a, RY20, BK21]), but it turns out not to be directly useful for our purposes as it is only known to hold for distributions satisfying very strong conditions such as rotational symmetry.

Limitations and Further Work. Open directions in testable learning (and universally testable learning) include the design of (efficient) tester-learners for concept classes other than the class of halfspaces, e.g., functions of halfspaces or neurons with other activations (like ReLU or sigmoid).

2.2 Preliminaries

Notation and Terminology. For what follows, we consider $D_{\mathcal{XY}}$ to be an unknown joint distribution over $\mathcal{X} \times \mathcal{Y}$ from which we receive independent samples, and its marginal on $\mathcal{X}$ will be denoted by $D_{\mathcal{X}}$. In particular $\mathcal{X} = \mathbb{R}^d$, and labels will lie in $\mathcal{Y} = \{\pm 1\}$. We will use $\mathcal{C}$ to denote a concept class mapping $\mathbb{R}^d$ to $\{\pm 1\}$, which throughout this chapter will be the class of halfspaces or functions of halfspaces over $\mathbb{R}^d$. We use $\mathrm{opt}(\mathcal{C}, D_{\mathcal{XY}})$ to denote the optimal error $\inf_{f \in \mathcal{C}} \mathbb{P}_{(\mathbf{x},y)\sim D_{\mathcal{XY}}}[f(\mathbf{x}) \neq y]$, or just opt when $\mathcal{C}$ and $D_{\mathcal{XY}}$ are clear from context. We recall

that in Massart noise model, the labels satisfy $\mathbb{P}_{y\sim D_{\mathcal{XY}}|\mathbf{x}}[y \neq \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle) \mid \mathbf{x}] = \eta(\mathbf{x})$, with $\eta(\mathbf{x}) \leq \eta < \frac{1}{2}$ for all $\mathbf{x}$. When we have adversarial noise (i.e., when we are in the agnostic model), the labels can be completely arbitrary. In both cases, the goal is to produce a hypothesis whose error is competitive with opt. We use $\mathbb{E}$ to denote the expectation of a random variable in brackets (or, correspondingly, $\mathbb{P}$ for the probability of an event), either over the unknown joint distribution or over the empirical distribution with respect to a sample S (e.g., $\mathbb{E}_{Z\in S}[f(Z)] = \frac{1}{|S|}\sum_{Z\in S} f(Z)$).

Definitions and Distributional Assumptions. For the problem of learning halfspaces in the agnostic and in Massart noise models, any of the known polynomial algorithms that achieve computationally optimal guarantees require that the marginal distribution has at least the following nice properties previously defined by, e.g., [DKTZ20b].

Definition 7 (Nice Distributions)**.** *For a given constant $\lambda \geq 1$, we consider the class of λ-nice distributions over $\mathbb{R}^d$ to be the distributions that satisfy the following properties:*

1. *For any unit vector $\mathbf{v}$ in $\mathbb{R}^d$ the distribution satisfies $\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2] \in [\frac{1}{\lambda}, \lambda]$.(bounded spectrum)*
2. *For any two dimensional subspace V, the corresponding marginal density $q_V(\mathbf{x})$ satisfies $q_V(\mathbf{x}) \geq 1/\lambda$ for any $\|\mathbf{x}\|_2 \leq 1/\lambda$. (anti-anti-concentration)*
3. *For any two dimensional subspace V, the corresponding marginal density $q_V(\mathbf{x})$ satisfies $q_V(\mathbf{x}) \leq Q(\|\mathbf{x}\|_2)$ for some function $Q : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\sup_{r\geq 0} Q(r) \leq \lambda$ and also $\int_{r=0}^{\infty} r^k Q(r)\, dr \leq \lambda$, for any $k = 1, 3, 5$. (anti-concentration and concentration)*

In the testable learning framework, however, corresponding results provide testable guarantees with respect to target marginals that are isotropic strongly log-concave , which is a strictly stronger condition than the one of Definition 7 (see Proposition 20 below). We now provide the standard definition of (strongly) log-concave distributions.

Definition 8 ((Strongly) Log-Concave Distributions [SW14])**.** *We say that a distribution over $\mathbb{R}^d$ is (β-strongly) log-concave, if its density can be written as $e^{-\varphi}$, where φ is a (β-strongly) convex function on $\mathbb{R}^d$ (for some $\beta > 0$).*

Proposition 20 (Log-Concave Distributions are Nice [LV07])**.** *There exists a universal constant $\lambda \geq 1$ such that any isotropic log-concave distribution is λ-nice.*

In this work, we provide universally testable guarantees with respect to the class of nice distributions with bounded Poincaré constant (see Definition 9 below).

Definition 9 (Poincaré Distributions)**.** *For a given value $\gamma > 0$, we say that a distribution over $\mathbb{R}^d$ is γ-Poincaré, if $\mathrm{var}(f(\mathbf{x})) \leq \gamma \cdot \mathbb{E}[\|\nabla f(\mathbf{x})\|_2^2]$ for any differentiable function $f : \mathbb{R}^d \to \mathbb{R}$.*

Although it is not clear whether one can efficiently obtain testable guarantees for the problem of learning noisy halfspaces under nice marginals (which is known to be an efficiently solvable problem in the non-testable setting [DKTZ20a, DKTZ20b]), by restricting our attention to nice distributions that, additionally, have bounded Poincaré constant, we obtain efficient learning results, even in the universally testable setting. Our results capture isotropic strongly log-concave distributions universally, due to Proposition 20 and the fact that strongly log-concave distributions are also Poincaré, as per Proposition 21 below.

Proposition 21 (Strongly Log-Concave Distributions are Poincaré, [SW14, Proposition 10.1])**.** *Any $\frac{1}{\gamma}$-strongly log-concave distribution is γ-Poincaré.*

Furthermore, under a long-standing conjecture about the geometry of convex bodies [KLS95], our results capture the family of all isotropic log-concave distributions.

Conjecture 10 (Kannan–Lovász–Simonovits Conjecture [KLS95] reformulation from [LV18])**.** *There is a universal constant $\gamma > 0$ for which any isotropic log-concave distribution is γ-Poincaré.*

2.3 Universal Testers

In this section, we present two basic testers that constitute the basic building blocks of the universal tester-learners we provide in the next section. The testers in this section might be of independent interest and their appeal is that they succeed even when the distribution in their input is unspecified up to certain bounds on a number of its statistics. In fact, the family of distributions for which each such tester succeeds is of infinite size, even non-parametric.

2.3.1 Universal Tester for Bounding Local Halfspace Disagreement

First, we present a universal tester that checks, given a parameter vector $\mathbf{w}$, whether a set of samples S is such that bounding the angular distance of $\mathbf{w}$ from an optimum parameter vector, implies that the corresponding halfspace disagrees with the (unknown) optimum halfspace only on a bounded fraction of points in S. This property ensures that if $\mathbf{w}$ is close to the optimum parameter vector, then it is also an approximate empirical risk minimizer. The tester universally accepts samples from nice distributions with high probability (Definition 7).

Lemma 6 (Universally Testable Bound for Local Halfspace Disagreement)**.** *Let $D_{\mathcal{XY}}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$, $\mathbf{w} \in \mathbb{S}^{d-1}$, $\theta \in (0, \pi/4]$, $\lambda \geq 1$ and $\delta \in (0,1)$. Then, for a sufficiently large constant C, there is a tester that given δ, θ, $\mathbf{w}$ and a set S of samples from $D_{\mathcal{X}}$ with size at least $C \cdot \left(\frac{d^4}{\theta^2 \delta}\right)$, runs in time $\mathrm{poly}\left(d, \frac{1}{\theta}, \frac{1}{\delta}\right)$ and satisfies the following specifications:*

1. *If the tester accepts S, then for every unit vector $\mathbf{w}' \in \mathbb{R}^n$ satisfying $\measuredangle(\mathbf{w}, \mathbf{w}') \leq \theta$ we have*

$$\mathbb{P}_{\mathbf{x} \sim S}[\mathit{sign}(\langle \mathbf{w}', \mathbf{x} \rangle) \neq \mathit{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)] \leq C \cdot \theta \cdot \lambda^C$$

2. *If the distribution $D_{\mathcal{X}}$ is λ-nice, the tester accepts S with probability $1 - \delta$.*

The proof of Lemma 6 exploits the observation that the probability of disagreement between two halfspaces can be upper bounded by a sum of products, where each product has two terms: one corresponding to the probability of falling in a (known) strip orthogonal to $\mathbf{w}$ and one corresponding to the probability of having large enough inner product with some unknown vector orthogonal to $\mathbf{w}$, conditioned in the (known) strip. We propose a tester that controls all of the

terms of the sum simultaneously by estimating the largest eigenvalue of a single covariance matrix (without conditioning). Upper and lower bounds on the eigenvalues of random symmetric matrices can be universally tested with testers that are guaranteed to accept when the elements of the matrix have bounded second moments (spectral tester of Proposition 26). We present our full proof in Appendix 2.6.1.

2.3.2 Universally Testable Weak Anti-Concentration

We now provide an important universal tester, which ensures that for a given vector $\mathbf{w}$, a sample set S and any unknown unit vector $\mathbf{v}$ orthogonal to $\mathbf{w}$, among the samples falling within a (known) strip orthogonal to $\mathbf{w}$, at least a constant fraction is absolutely correlated with $\mathbf{v}$ by a constant. In other words, the tester ensures that the conditional empirical distribution is weakly anti-concentrated in every direction. The tester universally accepts nice distributions that have bounded Poincaré constant.

Lemma 7 (Universally Testable Weak Anti-Concentration)**.** *Let D be a distribution over $\mathbb{R}^d$. Then, there is a universal constant $C > 0$ and a tester that given a unit vector $\mathbf{w} \in \mathbb{R}^d$, $\delta \in (0,1)$, $\gamma > 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{2\lambda}$ and a set S of i.i.d. samples from D with size at least $C \cdot \frac{d^4}{\sigma^2 \delta} \log(d) \lambda^C$, runs in time $\mathrm{poly}(d, \lambda, \frac{1}{\sigma}, \frac{1}{\delta}, \log\left(\frac{1}{\gamma}\right))$ and satisfies the following specifications*

1. *If the tester accepts S, then for any unit vector $\mathbf{v} \in \mathbb{R}^d$ with $\langle \mathbf{v}, \mathbf{w} \rangle = 0$ we have*

$$\mathop{\mathbb{P}}_{\mathbf{x} \in S}\left[|\langle \mathbf{v}, \mathbf{x} \rangle| \geq \frac{1}{C\lambda^C} \;\middle|\; |\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma \right] \geq \frac{1}{C\lambda^C \gamma^4}$$

2. *If D is γ-Poincaré and λ-nice, then the tester accepts S with probability at least $1 - \delta$.*

The proof of Lemma 7 is based on a simple fact from probability that is true for any non-negative random variable and ensures that the mass assigned to the tails is lower bounded by the ratio of the square of its expectation to the second moment.

Proposition 22 (Paley–Zygmund Inequality)**.** *For any non-negative random variable Z, we have*

$$\mathbb{P}[Z > \mathbb{E}[Z]/2] \geq \frac{1}{4} \cdot \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}$$

In the special case where Z follows the distribution of $\langle \mathbf{v}, \mathbf{x} \rangle^2$ conditioned on $|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma$ for some unitary orthogonal vectors $\mathbf{v}, \mathbf{w}$, some $\sigma > 0$ and some random variable $\mathbf{x}$ whose distribution is, say, 1-nice (see Definition 7), one can show that $\mathbb{E}[Z]$ is lower bounded by a constant and $\mathbb{E}[Z^2]$ is upper bounded by another constant, so Z assigns a non-trivial mass to a set that is bounded away from zero. This property is useful in the context of learning noisy halfspaces, as we show in the following section (see Proposition 25 and Lemma 8). However, testing algorithms that check whether such a property holds for given $\mathbf{w}$ and σ, are guaranteed to succeed when the marginal

distribution has, additionally, bounded Poincaré constant. The main part of the proof that requires a bounded Poincaré constant, is testing whether $\mathbb{E}[Z^2]$ is bounded uniformly over the set of unit vectors $\mathbf{v}$ orthogonal to $\mathbf{w}$, since $Z^2 = \langle \mathbf{v}, \mathbf{x} \rangle^4$, where $\mathbf{v}$ is unknown. We use the following result from [KS17].

Proposition 23 (Certifiable Hypercontractivity of Poincaré Distributions, Theorem 4.1 in [KS17])**.** *Let $\delta \in (0,1)$, $\gamma > 0$ and let D be a γ-Poincaré distribution over $\mathbb{R}^d$. Let S be a set of independent samples from D with size at least $(2d \log(4d/\delta))^4$. Consider the constrained maximization problem*

$$\arg \max_{\|\mathbf{v}\|_2 = 1} \mathbb{E}_{\mathbf{x} \in S} [\langle \mathbf{v}, \mathbf{x} \rangle^4] \tag{2.1}$$

Then, the optimum solution of the degree-4 sum-of-squares relaxation of the problem (2.1) *has value at most $C\gamma^4$ for some universal constant C, with probability at least $1 - \delta$ over the sample S.*

Using Proposition 23, we are able to provide a universal tester for bounding the empirical fourth moments. The tester solves an appropriate SDP relaxation of the (hard) problem [HL13] of finding the direction with maximum fourth moment and is guaranteed to succeed if $\mathbf{x}$ has Poincaré parameter bounded by a known value.

Proposition 24 (Hypercontractivity Tester)**.** *Let D be a distribution over $\mathbb{R}^d$. Then, there is a tester that given $\delta \in (0,1)$, $\gamma > 0$ and a set S of i.i.d. samples from D with size at least $(2d \log(4d/\delta))^4$, runs in time $\mathrm{poly}(d, \log \frac{1}{\delta}, \log \frac{1}{\gamma})$ and satisfies the following specifications*

1. *If the tester accepts S, then for any unit vector $\mathbf{v} \in \mathbb{R}^d$ we have*

$$\mathbb{E}_{\mathbf{x} \in S} [\langle \mathbf{v}, \mathbf{x} \rangle^4] \leq C \cdot \gamma^4 \,, \text{ where } C \text{ is some universal constant.}$$

2. *If the distribution D is γ-Poincaré, then the tester accepts S with probability at least $1 - \delta$.*

Proof. The tester does the following:

1. Solves a degree-4 sum-of-squares relaxation of problem (2.1) up to accuracy γ^4. (For a formal definition of the relaxed problem, see Problem (2.3) in [KS17].)

2. If the solution has value larger than $(C-1)\gamma^4$, then **reject**. Otherwise **accept**.

The computational complexity of the tester is $\mathrm{poly}(|S|, d, \log \frac{1}{\gamma})$, since the problem it solves can be written as a semidefinite program [Sho87, Par00, Nes00, Las01].

If the tester accepts S, then we know that the optimal solution of the relaxed problem is at most $C\gamma^4$ and we also know that any solution of the initial problem (2.1) has value at most equal to the value of the relaxation. Therefore $\mathbb{E}[\langle \mathbf{v}, \mathbf{x} \rangle^4] \leq C\gamma^4$, for any $\mathbf{v} \in \mathbb{S}^{d-1}$.

On the other hand, if the true distribution D is γ-Poincaré, then, with probability at least $1 - \delta$, we have that the solution found in step 2.3.2 has, with probability at least $1 - \delta$, value at most $C'\gamma^4$ for some universal constant C', due to Proposition 23. In order to ensure that the tester will accept with probability at least $1 - \delta$, it suffices to pick $C = C' + 1$. □

We provide the full proof of Lemma 7, in Appendix 2.6.2. The tests we perform include a spectral tester that accepts with high probability when the distribution of $\mathbf{x}$ is nice (similar to the spectral tester used for Lemma 6), a tester of the probability that $|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma$ and the hypercontractivity tester of Proposition 24.

2.4 Universal and Efficient Tester-Learners for Halfspaces

In this section, we present our main result on efficient and universal testable learning of halfspaces.

Theorem 11 (Efficient Universal Tester-Learner for Halfspaces)**.** *Let $D_{\mathcal{XY}}$ be any distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let $\mathcal{C}$ be the class of origin centered halfspaces in $\mathbb{R}^d$. Then, for any $\lambda \geq 1$, $\gamma > 0$, $\epsilon > 0$ and $\delta \in (0,1)$, there exists an universal tester-learner for $\mathcal{C}$ w.r.t. the class of λ-nice and γ-Poincaré marginals up to error $\mathrm{poly}(\lambda) \cdot (1+\gamma^4) \cdot \mathrm{opt} + \epsilon$, where $\mathrm{opt} = \min_{\mathbf{w} \in \mathbb{S}^{d-1}} \mathbb{P}_{D_{\mathcal{XY}}}[y \neq sign(\langle \mathbf{w}, \mathbf{x} \rangle)]$, and error probability at most δ, using a number of samples and running time $\mathrm{poly}(d, \lambda, \gamma, \frac{1}{\epsilon}, \log \frac{1}{\delta})$.*

Moreover, if the noise is Massart with given rate $\eta < 1/2$, then the algorithm achieves error $\mathrm{opt} + \epsilon$ with time and sample complexity $\mathrm{poly}(d, \lambda, \gamma, \frac{1}{\epsilon}, \frac{1}{1-2\eta}, \log \frac{1}{\delta})$.

Our proof follows a surrogate loss minimization approach that has been used for classical learning of noisy halfspaces [DKTZ20a, DKTZ20b]. In particular, the algorithm runs Projected Stochastic Gradient Descent (see 29) on a surrogate loss whose stationary points are shown to be close to optimum parameter vectors under certain distributional assumptions.

We use the following surrogate loss function.

$$\mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{XY}}) = \mathop{\mathbb{E}}_{(\mathbf{x},y) \sim D_{\mathcal{XY}}} \left[\ell_\sigma \left(-y \frac{\langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{w}\|_2} \right) \right], \tag{2.2}$$

In Equation (2.2), the function ℓ_σ is a smoothed version of the step function as in Proposition 28.

In order to analyze the properties of the stationary points of the surrogate loss, we provide the following refinement of results implicit in [DKTZ20a, DKTZ20b]. We show that the gradient of the surrogate loss is lower bounded by the difference between certain quantities that are controlled by the marginal distribution (see Figure 2-2). We stress that we do not use any assumptions for the marginal distribution in this step. Prior work included similar bounds, but the corresponding quantities were different. We need to be more precise and provide the following result, whose proof is based on two dimensional geometry and can be found in Appendix 2.7.1.

Proposition 25 (Modification from [DKTZ20a, DKTZ20b])**.** *For a distribution $D_{\mathcal{XY}}$ over $\mathbb{R}^d \times \{\pm 1\}$ let opt be the minimum error achieved by some origin-centered halfspace and $\mathbf{w}^* \in \mathbb{S}^{d-1}$ a corresponding vector. Consider $\mathcal{L}_\sigma$ as in Equation (2.2) for $\sigma > 0$ and let $\eta < 1/2$. Let $\mathbf{w} \in \mathbb{S}^{d-1}$ with $\measuredangle(\mathbf{w}, \mathbf{w}^*) = \theta < \frac{\pi}{2}$ and $\mathbf{v} \in \mathrm{span}(\mathbf{w}, \mathbf{w}^*)$ such that $\langle \mathbf{v}, \mathbf{w} \rangle = 0$ and $\langle \mathbf{v}, \mathbf{w}^* \rangle < 0$. Then, for some universal constant $C > 0$ and any $\alpha \geq \frac{\sigma}{2 \tan \theta}$ we have $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{XY}})\|_2 \geq A_1 - A_2 - A_3$,*

where

$$A_1 = \frac{\alpha}{C \cdot \sigma} \cdot \mathbb{P}\left[|\langle \mathbf{v}, \mathbf{x}\rangle| \geq \alpha \text{ and } |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}\right]$$
$$A_2 = \frac{C}{\tan\theta} \cdot \mathbb{P}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}\right] \text{ and } A_3 = \frac{C}{\sigma} \cdot \sqrt{\mathrm{opt}} \cdot \sqrt{\mathbb{E}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}_{\{|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}\}}\right]}$$

Moreover, if the noise is Massart with rate η, then $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{XY}})\|_2 \geq (1-2\eta)A_1 - A_2$.

If the marginal distribution is nice, then the quantities A_1, A_2 and A_3 are such that σ can be chosen accordingly so that stationary points of the surrogate loss (or their inverses) are close to some optimum vector (see Proposition 27 for properties of nice distributions). We use some simple tests (e.g., estimate the probability of falling in a strip, $\mathbb{P}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma/2]$ and appropriate spectral testers) as well as our universal tester for weak anti-concentration (see 7) to establish bounds on quantities A_1, A_2 and A_3 which ensure that the desired property holds for a given vector $\mathbf{w}$, under no distributional assumptions. The tester in the following result universally accepts nice distributions with bounded Poincaré parameter. The formal proof can be found in Appendix 2.7.2.

Lemma 8 (Universally Testable Structure of Surrogate Loss)**.** *Let $D_{\mathcal{XY}}$ be any distribution over $\mathbb{R}^d \times \{\pm 1\}$. Consider $\mathcal{L}_\sigma$ as in Equation* (2.2)*. Then, there is a universal constant $C > 0$ and a tester that given a unit vector $\mathbf{w} \in \mathbb{R}^d$, $\delta \in (0,1)$, $\eta < 1/2$, $\gamma > 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{C\lambda^C}$ and a set S of i.i.d. samples from $D_{\mathcal{XY}}$ with size at least $C \cdot \frac{d^4}{\sigma^2\delta}\log(d)\lambda^C$, runs in time $\mathrm{poly}(d, \lambda, \frac{1}{\sigma}, \frac{1}{\delta}, \log\left(\frac{1}{\gamma}\right))$ and satisfies the following specifications*

1. *If the tester accepts S, then, the following statements are true for the minimum error opt_S achieved by some origin-centered halfspace on S and the optimum vector $\mathbf{w}_S^* \in \mathbb{S}^{d-1}$*
   - *If the noise is Massart with associated rate η and $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \leq \frac{1-2\eta}{C\lambda^C\gamma^4}$ then either $\measuredangle(\mathbf{w}, \mathbf{w}_S^*) \leq \frac{C\lambda^C(1+\gamma^4)}{1-2\eta} \cdot \sigma$ or $\measuredangle(-\mathbf{w}, \mathbf{w}_S^*) \leq \frac{C\lambda^C(1+\gamma^4)}{1-2\eta} \cdot \sigma$.*
   - *If the noise is adversarial with $\mathrm{opt}_S \leq \frac{\sigma}{C\lambda^C}$ and $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S)\|_2 < \frac{1}{C\lambda^C\gamma^4}$ then either $\measuredangle(\mathbf{w}, \mathbf{w}_S^*) \leq C\lambda^C(1+\gamma^4) \cdot \sigma$ or $\measuredangle(-\mathbf{w}, \mathbf{w}_S^*) \leq C\lambda^C(1+\gamma^4) \cdot \sigma$.*
2. *If the marginal $D_{\mathcal{X}}$ is λ-nice and γ-Poincaré, then the tester accepts S with probability at least $1-\delta$.*

We now give the algorithm for $\delta \leftarrow 1/3$ since we can reduce the probability of failure with repetition (repeat $O(\log \frac{1}{\delta})$ times, accept if the rate of acceptance is $\Omega(1)$ and output the halfspace achieving the minimum test error among the halfspaces returned).

The algorithm receives $\lambda \geq 1$, $\gamma > 0$, $\epsilon > 0$ and $\eta \in (0, 1/2) \cup \{1\}$ (say $\eta = 1$ when we are in the agnostic case) and does the following for some appropriately large universal constants $C_1, C_2 > 0$.

1. First, initialize $E = \frac{\epsilon}{C_1\lambda^{C_1}}$, and let Σ be a list of real numbers and A be a positive real number, where Σ and A are defined as follows. If $\eta = 1$, then Σ is an $\frac{E}{C_1\lambda^{C_1}}$-cover of the interval $\left[0, \frac{1}{C_1\lambda^{C_1}}\right]$ and $A = \frac{1}{C_1\lambda^{C_1}\gamma^4}$. Otherwise, let $\Sigma = \left\{\frac{E\cdot(1-2\eta)}{C_1\lambda^{C_1}(1+\gamma^4)}\right\}$ and $A = \frac{1-2\eta}{C_1\lambda^{C_1}\gamma^4}$.

2. Draw a set S_1 of $C_2 \left(\frac{\lambda d}{\gamma\epsilon}\right)^{C_2}$ i.i.d. samples from $D_{\mathcal{X}\mathcal{Y}}$ and run PSGD, as specified in Proposition 29 with $\epsilon \leftarrow A$, $\delta \leftarrow \frac{\delta}{C_1}$ on the loss $\mathcal{L}_\sigma$ for each $\sigma \in \Sigma$.

3. Form a list L with all the pairs of the form $(\mathbf{w}, \sigma)$ where $\mathbf{w} \in \mathbb{S}^{d-1}$ is some iterate of the PSGD subroutine performed on $\mathcal{L}_\sigma$.

4. Draw a fresh set S_2 of $C_2 \left(\frac{\lambda d}{\gamma\epsilon}\right)^{C_2}$ i.i.d. samples from $D_{\mathcal{X}\mathcal{Y}}$ and compute for each $(\mathbf{w}, \sigma) \in L$ the value $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2$. If, for some $\sigma \in \Sigma$, $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2 > A$ for all $(\mathbf{w}, \sigma) \in L$, then **reject**.

5. Update L by keeping for each $\sigma \in \Sigma$ only one pair of the form $(\mathbf{w}, \sigma)$ for which we have $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2 \leq A$.

6. Run the following tests for each $(\mathbf{w}, \sigma) \in L$. (This will ensure that part (a) of Lemma 8 holds for each of the elements of L, i.e., that any stationary point of the loss $\mathcal{L}_\sigma$ that lies in L is angularly close to the empirical risk minimizer[3].).

   - If $\mathbb{P}_{(\mathbf{x},y)\in S_2}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}] \leq \frac{\sigma}{C_1\lambda^{C_1}}$ or $\mathbb{P}_{(\mathbf{x},y)\in S_2}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}] > \sigma \cdot C_1\lambda^{C_1}$, then **reject**.
   - Compute the $(d-1) \times (d-1)$ matrices $M^+_{S_2}$ and $M^-_{S_2}$ as follows:[4]

$$M^+_{S_2} = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})^T \cdot \mathbb{1}_{\{|\langle \mathbf{w},\mathbf{x}\rangle| \leq \frac{\sigma}{2}\}}\right]$$
$$M^-_{S_2} = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})^T \cdot \mathbb{1}_{\{|\langle \mathbf{w},\mathbf{x}\rangle| \leq \frac{\sigma}{6}\}}\right]$$

   - **Reject** if the maximum singular value of $M^+_{S_2}$ is greater than $\sigma \cdot C_1\lambda^{C_1}$.
   - **Reject** if the minimum singular value of $M^-_{S_2}$ is less than $\frac{\sigma}{C_1\lambda^{C_1}}$.
   - Run the hypercontractivity tester on $S' = \{\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x} : (\mathbf{x}, y) \in S_2 \text{ and } |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}$, i.e., solve an appropriate SDP (see Prop. 24 with $\gamma \leftarrow \gamma$, $\delta \leftarrow \delta/C_1$) and **reject** if the solution is larger than a specified threshold.

7. Set $\theta = \frac{(1+\gamma^4)\sigma}{A\gamma^4}$, and run the following tests for each pair of the form $(\mathbf{w}, \sigma)$ and $(-\mathbf{w}, \sigma)$ where $(\mathbf{w}, \sigma) \in L$. (This will ensure that part (a) of Lemma 6 is activated, i.e., that the distance of a vector from the empirical risk minimizer is an accurate proxy for the error of the corresponding halfspace.)

   - If $\mathbb{P}_{(\mathbf{x},y)\in S_2}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \theta] > C_1\lambda^{C_1}\theta$ then **reject**.

[3] Or the same holds for the inverse vector.

[4] The operator $\mathrm{proj}_{\perp\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}^{d-1}$ projects vectors on the hyperplane orthogonal to $\mathbf{w}$.

- Compute the $(d-1) \times (d-1)$ matrix M_{S_2} as follows:[5]

$$M_{S_2} = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[\sum_{i=2}^{\infty} \frac{(\mathrm{proj}_{\perp \mathbf{w}}\,\mathbf{x})(\mathrm{proj}_{\perp \mathbf{w}}\,\mathbf{x})^T}{(i-1)^2}\, \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \in [(i-1)\theta, i\theta)\}\right]$$

- If $\|M_S\|_{\mathrm{op}} > C_1\theta\lambda^{C_1}$, then **reject**.

8. Otherwise, **accept** and output the vector $\mathbf{w}$ that achieves the smallest empirical error on S_2 among the vectors in the list L.

This concludes the algorithm. The full proof of Theorem 11 may be found in Appendix 2.7.3.

2.5 Technical Lemmas

In this section, we provide a list of technical results that we use in our proofs.

Lemma 9 (Preservation of Poincaré constant)**.** *Let I be an open interval in $\mathbb{R}$ and $q : \mathbb{R}^d \to \mathbb{R}_+$ the density of a γ-Poincaré distribution. Let $\mathbf{v} \in \mathbb{S}^{d-1}$ and $q'_{\mathbf{v}} : \mathbb{R}^{d-1} \to \mathbb{R}_+$ be the density of the distribution resulting from conditioning q to $\mathbf{x}\cdot\mathbf{v} \in I$ and projecting on the subspace perpendicular to $\mathbf{v}$. Then, the distribution corresponding to $q'_{\mathbf{v}}$ is γ-Poincaré.*

Proof. Assume, without loss of generality, that $\mathbf{v} = \mathbf{e}_d$. We have that

$$q'_{\mathbf{v}}(\mathbf{x}_{<d}) = \frac{\int_{x_d\in I} q(\mathbf{x}_{<d}, x_d)\, dx_d}{\int_{\mathbf{x}_{<d}}\int_{x_d\in I} q(\mathbf{x})\, d\mathbf{x}}\,, \text{ for any } \mathbf{x}_{<d} \in \mathbb{R}^{d-1}\,.$$

Let $f : \mathbb{R}^{d-1} \to \mathbb{R}$ be any differentiable function. In order to show that $q'_{\mathbf{v}}$ is γ-Poincaré, it is sufficient to show that under no further assumptions on f, the quantity $\mathrm{var}_{q'_{\mathbf{v}}}(f(\mathbf{x}_{<d}))$ is upper bounded by the product of γ and $\mathbb{E}_{q'_{\mathbf{v}}}[\|\nabla f(\mathbf{x}_{<d})\|_2^2]$. We expand the quantity $\mathrm{var}_{q'_{\mathbf{v}}}(f(\mathbf{x}_{<d}))$ as

[5]Note that only at most $|S_2|$ many terms below are non-zero, hence M_{S_2} can be computed efficiently.

follows

$$\begin{aligned}
\mathrm{var}_{q'_{\mathbf{v}}}(f(\mathbf{x}_{<d})) &= \inf_{\tau} \int_{\mathbf{x}_{<d}} (f(\mathbf{x}_{<d}) - \tau)^2 q'_{\mathbf{v}}(\mathbf{x}_{<d})\, d\mathbf{x}_{<d} \\
&= \inf_{\tau} \int_{\mathbf{x}_{<d}} (f(\mathbf{x}_{<d}) - \tau)^2 \cdot \frac{\int_{x_d \in I} q(\mathbf{x}_{<d}, x_d)\, dx_d}{\int_{\mathbf{x}_{<d}} \int_{x_d \in I} q(\mathbf{x})\, d\mathbf{x}}\, d\mathbf{x}_{<d} \\
&= \frac{\inf_{\tau} \int_{\mathbf{x}_{<d}} \int_{x_d \in I} (f(\mathbf{x}_{<d}) - \tau)^2 \cdot q(\mathbf{x})\, dx_d\, d\mathbf{x}_{<d}}{\int_{\mathbf{x}_{<d}} \int_{x_d \in I} q(\mathbf{x})\, d\mathbf{x}} \\
&\leq \frac{\gamma \cdot \int_{\mathbf{x}_{<d}} \int_{x_d \in I} \|\nabla_{\mathbf{x}} f(\mathbf{x}_{<d})\|_2^2 \cdot q(\mathbf{x})\, d\mathbf{x}}{\int_{\mathbf{x}_{<d}} \int_{x_d \in I} q(\mathbf{x})\, d\mathbf{x}} && \text{(since } q \text{ is } \gamma\text{-Poincaré)} \\
&= \gamma \cdot \int_{\mathbf{x}_{<d}} \|\nabla_{\mathbf{x}_{<d}} f(\mathbf{x}_{<d})\|_2^2 \cdot q'_{\mathbf{v}}(\mathbf{x}_{<d})\, d\mathbf{x}_{<d} && \text{(since } \tfrac{\partial f}{\partial x_d} \equiv 0) \\
&= \gamma \cdot \mathop{\mathbb{E}}_{q'_{\mathbf{v}}}[\|\nabla f(\mathbf{x}_{<d})\|_2^2]\,,
\end{aligned}$$

which concludes the proof. □

Proposition 26 (Spectral Tester)**.** *Let D be a distribution over $\mathbb{R}^d$. Then, there is a tester that given $\delta \in (0,1)$, $\lambda \geq 1$, $\theta > 0$ and a set S of i.i.d. samples from D with size at least $\frac{2\lambda d^4}{\theta^2 \delta}$, runs in time $\mathrm{poly}(d, \frac{1}{\theta}, |S|)$ and satisfies the following specifications*

1. *If the tester accepts, then, for $\mathbf{z} \sim S$, $\mathbb{E}_S[\mathbf{z}\mathbf{z}^T] \succeq \frac{\theta}{2} I_d$ (resp. $\mathbb{E}_S[\mathbf{z}\mathbf{z}^T] \preceq 2\theta I_d$).*
2. *If, for $\mathbf{z} \sim D$, $\mathbb{E}_D[(\mathbf{z}_i \mathbf{z}_j)^2] \leq \lambda$ and $\mathbb{E}_D[\mathbf{z}\mathbf{z}^T] \succeq \theta I_d$ (resp. $\mathbb{E}_D[\mathbf{z}\mathbf{z}^T] \preceq \theta I_d$), then the tester accepts with probability at least $1 - \delta$.*

Proof. The tester receives λ, a set S and $\delta \in (0,1)$ and does the following:

1. Compute the matrix $M_S = \mathbb{E}_S[\mathbf{z}\mathbf{z}^T]$.
2. If the minimum (resp. maximum) eigenvalue of M_S is larger than $\frac{\theta}{2}$ (resp. smaller than 2θ), then **accept**. Otherwise **reject**.

Clearly, if the tester accepts, then the desired property is satisfied by construction. If the distribution D satisfies the conditions of part 2, we can show that for $M_D = \mathbb{E}_{\mathbf{z} \sim D}[\mathbf{z}\mathbf{z}^T]$ we have

$$\left\| M_S - M_D \right\|_{\mathrm{op}} \leq \frac{\theta}{2}, \text{ with probability at least } 1 - \delta$$

which implies that $M_S \succeq \frac{\theta}{2} I_d$ (and $M_S \preceq (\theta + \frac{\theta}{2}) I_d \preceq 2\theta I_d$). In particular, we have that $(M_S)_{ij} = \mathbb{E}_S[\mathbf{z}_i \mathbf{z}_j]$, and by Chebyshev's inequality we have

$$\mathbb{P}\left[|(M_S)_{ij} - (M_D)_{ij}| > \frac{\theta}{2d} \right] \leq \frac{4d^2}{\theta^2 |S|} \mathop{\mathbb{E}}_{\mathbf{z} \sim D}[(\mathbf{z}_i \mathbf{z}_j)^2] \leq \frac{4\lambda d^2}{\theta^2 |S|} \leq \frac{\delta}{\binom{d}{2}}$$

By a union bound, we get that $\|M_S - M_D\|_{\max} \leq \frac{\theta}{2d}$ with probability at least $1 - \delta$ and hence $\|M_S - M_D\|_{\text{op}} \leq d\|M_S - M_D\|_{\max} \leq \frac{\theta}{2}$, which concludes the proof. □

Proposition 27. *Let $c \geq 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{2\lambda}$ and D be a λ-nice distribution over $\mathbb{R}^d$. Then, for any unit vectors $\mathbf{w}, \mathbf{v}, \mathbf{v}', \mathbf{u}, \mathbf{u}' \in \mathbb{R}^d$ with $\langle \mathbf{w}, \mathbf{v} \rangle = \langle \mathbf{w}, \mathbf{v}' \rangle = 0$ and for some universal constant $C > 0$ we have*

(i) $\mathbb{P}[|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma] = 2\sigma \cdot \alpha^C$, *for some* $\alpha \in [\frac{1}{C\lambda}, C\lambda]$.

(ii) $\mathbb{E}[\langle \mathbf{v}, \mathbf{x} \rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma\}] = 2\sigma \cdot \alpha^C$, *for some* $\alpha \in [\frac{1}{C\lambda}, C\lambda]$.

(iii) $\mathbb{E}[\langle \mathbf{x}, \mathbf{u} \rangle^2 \langle \mathbf{x}, \mathbf{u}' \rangle^2] = \alpha^C$, *for some* $\alpha \leq C\lambda$.

(iv) $\mathbb{E}[\langle \mathbf{v}, \mathbf{x} \rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x} \rangle| \in [c, c + \sigma]\}] \leq 2\sigma \cdot \alpha^C$, *for some* $\alpha \leq C\lambda$.

Proof. We start by deriving property (i). Recall the function Q from the definition of a λ-nice distribution, which upper-bounds the density of any two-dimensional projection of a λ-nice distribution we see that:

$$\begin{aligned}\mathbb{P}[|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma] &= \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2 \\ &\leq \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} Q\left(\sqrt{x_1^2 + x_2^2}\right) dx_1 dx_2\end{aligned}$$

Now, note that the region $\{(x_1, x_2) : |x_1| \leq \sigma\}$ is a subset of the set

$$\{(x_1, x_2) : |x_2| \leq \sigma |x_1|\} \cup \{(x_1, x_2) : |x_1| \leq \sigma \;\&\; |x_2| \leq 1\}.$$

Therefore:

$$\begin{aligned}&\int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} Q\left(\sqrt{x_1^2 + x_2^2}\right) dx_1 dx_2 \leq \\ &\qquad 4 \arcsin(\sigma) \cdot \int_{r=0}^{\infty} 2\pi r Q\left(r\right)\, dr + \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-1}^{1} Q\left(\sqrt{x_1^2 + x_2^2}\right) dx_1 dx_2 \leq O(\sigma\lambda)\end{aligned}$$

Note that in the last line above, we bounded the first term via the bound $\int_{r=0}^{\infty} rQ\left(r\right)\, dr \leq \lambda$ from the definition of λ-nice distributions. Likewise, we bounded the second term via the inequality $Q(r) \leq \lambda$ from the definition of λ-nice distributions. Overall, we get

$$\mathbb{E}[\langle \mathbf{v}, \mathbf{x} \rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \sigma\}] \leq O(\sigma\lambda)$$

Now, we shall lower-bound the same quantity. We have

$$\begin{aligned}\mathbb{P}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma] &= \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2 \\ &\geq \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\frac{1}{2\lambda}}^{\frac{1}{2\lambda}} q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2\end{aligned}$$

Now, since $\sigma \leq \frac{1}{2\lambda}$ via the premise of the lemma, we see that the whole region of integration on the right side of the set $\{(x_1, x_2) : \sqrt{x_1^2 + x_2^2} \leq \frac{1}{\lambda}\}$. From the definition of λ-nice distributions, the density $q_{\text{span}(\mathbf{v},\mathbf{w})}$ is lower-bounded by $1/\lambda$ in this region. Therefore, we have

$$\mathbb{P}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma] \geq \frac{2\sigma}{\lambda} \cdot \frac{1}{\lambda} = \frac{2\sigma}{\lambda^2},$$

which finishes the proof of property (i).

Now, we derive property (ii). Recall the function Q from the definition of a λ-nice distribution, which upper-bounds the density of any two-dimensional projection of a λ-nice distribution we see that:

$$\begin{aligned}\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}] &= \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2 \\ &\leq \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot Q\left(\sqrt{x_1^2 + x_2^2}\right)\, dx_1 dx_2\end{aligned}$$

Now, note that the region $\{(x_1, x_2) : |x_1| \leq \sigma\}$ is a subset of the set

$$\{(x_1, x_2) : |x_2| \leq \sigma |x_1|\} \cup \{(x_1, x_2) : |x_1| \leq \sigma \;\&\; |x_2| \leq 1\}.$$

Therefore:

$$\begin{aligned}&\int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot Q\left(\sqrt{x_1^2 + x_2^2}\right)\, dx_1 dx_2 \leq \\ &\quad 4 \arcsin(\sigma) \cdot \int_{r=0}^{\infty} 2\pi r^3 Q\left(r\right)\, dr + \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-1}^{1} x_2^2 \cdot Q\left(\sqrt{x_1^2 + x_2^2}\right)\, dx_1 dx_2 \leq O(\sigma\lambda)\end{aligned}$$

Note that in the last line above, we bounded the first term via the bound on $\int_{r=0}^{\infty} r^3 Q\left(r\right)\, dr$ from the definition of λ-nice distributions. Likewise, we bounded the second term via the inequality $Q(r) \leq \lambda$ from the definition of λ-nice distributions. Therefore, we get

$$\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}] \leq O(\sigma\lambda)$$

Now, we shall lower-bound the same quantity. We have

$$\begin{aligned}\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \le \sigma\}] &= \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2 \\ &\ge \int_{x_1=-\sigma}^{\sigma} \int_{x_2=-\frac{1}{2\lambda}}^{\frac{1}{2\lambda}} x_2^2 \cdot q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2\end{aligned}$$

Now, since $\sigma \le \frac{1}{2\lambda}$ via the premise of the lemma, we see that the whole region of integration on the right side of the set $\{(x_1, x_2) : \sqrt{x_1^2 + x_2^2} \le \frac{1}{\lambda}\}$. From the definition of λ-nice distributions, the density $q_{\text{span}(\mathbf{v},\mathbf{w})}$ is lower-bounded by $1/\lambda$ in this region. Therefore, we have

$$\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \le \sigma\}] \ge \frac{2\sigma}{\lambda} \cdot \frac{1}{4\lambda^2} \cdot \frac{1}{\lambda} = \frac{\sigma}{2\lambda^4},$$

which finishes the proof of property (ii).

We proceed to property (iii). We will denote the angle between $\mathbf{v}$ and $\mathbf{v}'$ as β, which allows us to write

$$\begin{aligned}\mathbb{E}[\langle \mathbf{x}, \mathbf{v}\rangle^2 \langle \mathbf{x}, \mathbf{v}'\rangle^2] &= \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} x_1^2 (x_1 \cos\beta + x_2 \sin\beta)^2 q_{\text{span}(\mathbf{v},\mathbf{w})}(x_1, x_2)\, dx_1 dx_2 \\ &\le \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} x_1^2 (x_1 \cos\beta + x_2 \sin\beta)^2 \cdot Q\left(\sqrt{x_1^2 + x_2^2}\right) dx_1 dx_2 \\ &\le \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} (x_1^2 + x_2^2)^2 \cdot Q\left(\sqrt{x_1^2 + x_2^2}\right) dx_1 dx_2 \\ &= \int_{r=0}^{\infty} 2\pi r^5 Q(r)\, dr \le 2\pi\lambda,\end{aligned}$$

which finishes the proof of property (iii).

Finally, we prove property (iv). For $\beta \ge 0$ we have

$$\begin{aligned}\int_{r=0}^{\infty} r^2 Q\big(\sqrt{r^2+\beta}\big)\, dr &= \int_{r=0}^{1} r^2 Q\big(\sqrt{r^2+\beta}\big)\, dr + \int_{r=1}^{\infty} r^2 Q\big(\sqrt{r^2+\beta}\big)\, dr \\ &\le \lambda + \int_{r=1}^{\infty} r^3 Q\big(\sqrt{r^2+\beta}\big)\, dr && \text{(since } \sup_{r\ge 0} Q(r) \le \lambda\text{)} \\ &\le \lambda + \int_{r'=\sqrt{1+\beta}}^{\infty} (r'^3 - \beta r') Q(r')\, dr' && \text{(by setting } r' = \sqrt{r^2+\beta}\text{)} \\ &\le \lambda + \int_{r=0}^{\infty} r^3 Q(r)\, dr && \text{(since } \beta r Q(r) \ge 0 \text{ for any } r \ge 0\text{)} \\ &\le 2\lambda\end{aligned}$$

Applying the above inequality to the quantity of property (iv), we get the desired result.

$$\begin{aligned}
\mathbb{E}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \in [c, c+\sigma]\}] &= \int_{|x_1|\in[c,c+\sigma]} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot q_{\mathrm{span}(\mathbf{v},\mathbf{w})}\, dx_1 dx_2 \\
&\leq \int_{|x_1|\in[c,c+\sigma]} \int_{x_2=-\infty}^{\infty} x_2^2 \cdot Q\big(\sqrt{x_1^2+x_2^2}\big)\, dx_1 dx_2 \\
&= \int_{|x_1|\in[c,c+\sigma]} \left(2\int_{r=0}^{\infty} r^2 \cdot Q\big(\sqrt{x_1^2+r^2}\big)\, dr\right) dx_1 \\
&\leq \int_{|x_1|\in[c,c+\sigma]} (4\lambda)\, dx_1 \leq 8\lambda\sigma
\end{aligned}$$

This concludes the proof of Proposition 27. □

Proposition 28. *There is a universal constant $C > 0$, such that for any $\sigma > 0$, there exists a continuously differentiable function $\ell_\sigma : \mathbb{R} \to [0,1]$ with the following properties.*

1. *For any $t \in [-\sigma/6, \sigma/6]$, $\ell_\sigma(t) = \frac{1}{2} + \frac{t}{\sigma}$.*
2. *For any $t > \sigma/2$, $\ell_\sigma(t) = 1$ and for any $t < -\sigma/2$, $\ell_\sigma(t) = 0$.*
3. *For any $t \in \mathbb{R}$, $\ell'_\sigma(t) \in [0, C/\sigma]$, $\ell'_\sigma(t) = \ell'_\sigma(-t)$ and $|\ell''_\sigma(t)| \leq C/\sigma^2$.*

Proof. We define ℓ_σ as follows.

$$\ell_\sigma(t) = \begin{cases} \frac{t}{\sigma} + \frac{1}{2}, \text{ if } |t| \leq \frac{\sigma}{6} \\ 1, \text{ if } t > \frac{\sigma}{2} \\ 0, \text{ if } t < \frac{-\sigma}{2} \\ \ell^+(t), t \in (\frac{\sigma}{6}, \frac{\sigma}{2}] \\ \ell^-(t), t \in [-\frac{\sigma}{2}, -\frac{\sigma}{6}) \end{cases}$$

for some appropriate functions ℓ^+, ℓ^-. It is sufficient that we pick ℓ^+ satisfying the following conditions (then ℓ^- would be defined symmetrically, i.e., $\ell^-(t) = 1 - \ell^+(-t)$).

- $\ell^+(\sigma/2) = 1$ and $\ell^{+\prime}(\sigma/2) = 0$.
- $\ell^+(\sigma/6) = 2/3$ and $\ell^{+\prime}(\sigma/6) = 1/\sigma$.
- $\ell^{+\prime\prime}$ is defined and bounded, except, possibly on $\sigma/6$ and/or $\sigma/2$.

We therefore need to satisfy four equations for ℓ^+. So we set ℓ^+ to be a degree 3 polynomial: $\ell^+(t) = a_1 t^3 + a_2 t^2 + a_3 t + a_4$. Whenever $\sigma > 0$, the system has a unique solution that satisfies the desired inequalities. In particular, we may solve the equation to get $a_1 = -9/\sigma^3, a_2 = 15/(2\sigma^2), a_3 = -3/(4\sigma)$ and $a_4 = 5/8$. For the resulting function (see Figure 2-1 below) we

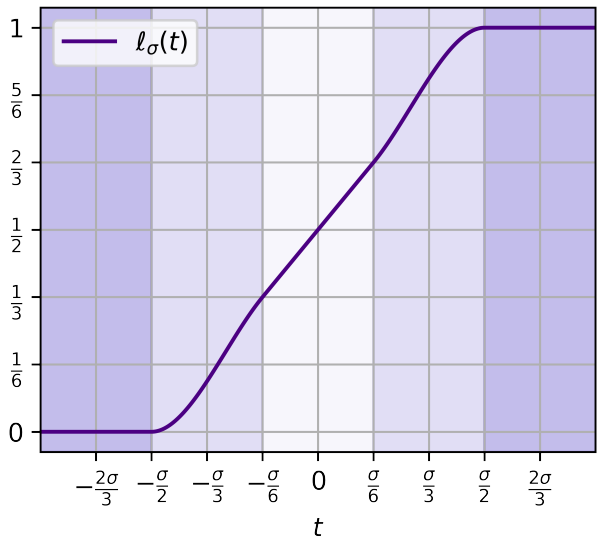

Figure 2-1: The function ℓ_σ used to smoothly approximate the ramp.

have that there are constants $c, c' > 0$ such that $\ell^{+\prime}(t) \in [0, c/\sigma]$ and $|\ell^{+\prime\prime}(t)| \leq c'/\sigma^2$ for any $t \in [\sigma/6, \sigma/2]$.

□

Proposition 29 (PSGD Convergence [DKTZ20a])**.** *Let $\mathcal{L}_\sigma$ be as in Equation* (2.2) *with $\sigma \in (0, 1]$, ℓ_σ as described in Proposition 28, $\lambda \geq 1$ and $D_{\mathcal{XY}}$ such that the marginal $D_{\mathcal{X}}$ on $\mathbb{R}^d$ is λ-nice. Then for some universal constant $C > 0$ and for any $\epsilon > 0$ and $\delta \in (0, 1)$, there is an algorithm whose time and sample complexity is $O(\frac{\lambda^C d}{\sigma^4} + \frac{\lambda^C \log(1/\delta)}{\epsilon^4 \sigma^4})$, which, having access to samples from $D_{\mathcal{XY}}$, outputs a list L of vectors $\mathbf{w} \in \mathbb{S}^{d-1}$ with $|L| = O(\frac{\lambda^C d}{\sigma^4} + \frac{\lambda^C \log(1/\delta)}{\epsilon^4 \sigma^4})$ so that there exists $\mathbf{w} \in L$ with*

$$\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{XY}})\|_2 \leq \epsilon\,, \text{ with probability at least } 1 - \delta\,.$$

In particular, the algorithm performs Stochastic Gradient Descent on $\mathcal{L}_\sigma$ Projected on $\mathbb{S}^{d-1}$ (PSGD).

2.6 Proofs from Section 2.3

2.6.1 Proof of Lemma 6

We restate Lemma 6 here for convenience.

Lemma 10 (Lemma 6)**.** *Let $D_{\mathcal{XY}}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$, $\mathbf{w} \in \mathbb{S}^{d-1}$, $\theta \in (0, \pi/4]$, $\lambda \geq 1$ and $\delta \in (0, 1)$. Then, for a sufficiently large constant C, there is a tester that given δ, θ, $\mathbf{w}$ and a set S of samples from $D_{\mathcal{X}}$ with size at least $C \cdot \left(\frac{d^4}{\theta^2 \delta}\right)$, runs in time $\mathrm{poly}\left(d, \frac{1}{\theta}, \frac{1}{\delta}\right)$ and satisfies the following specifications:*

(*a*) *If the tester accepts S, then for every unit vector $\mathbf{w}' \in \mathbb{R}^n$ satisfying $\measuredangle(\mathbf{w}, \mathbf{w}') \leq \theta$ we have*

$$\mathbb{P}_{\mathbf{x}\sim S}[\mathit{sign}(\langle \mathbf{w}', \mathbf{x}\rangle) \neq \mathit{sign}(\langle \mathbf{w}, \mathbf{x}\rangle)] \leq C \cdot \theta \cdot \lambda^C$$

(*b*) *If the distribution $D_{\mathcal{X}}$ is λ-nice, the tester accepts S with probability $1 - \delta$.*

Proof. The testing algorithm receives integer d, set $S \subset \mathbb{R}^d$, $\mathbf{w} \in \mathbb{S}^{d-1}$, $\theta \in (0, \pi/4]$, $\lambda \geq 1$ and $\delta \in (0, 1)$ and does the following for some sufficiently large universal constant $C_1 > 0$:

1. If $\mathbb{P}_{\mathbf{x}\in S}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \in [0, \theta]\right] > C_1\theta\lambda^{C_1}$, then **reject**.
2. Let $\mathrm{proj}_{\perp\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}^{d-1}$ denote the operator that given any vector in $\mathbb{R}^d$, it outputs its projection into the $(d-1)$-dimensional subspace of $\mathbb{R}^d$ that is orthogonal to $\mathbf{w}$.
3. Compute the $(d-1) \times (d-1)$ matrix M_S as follows[6]:

$$M_S = \mathop{\mathbb{E}}_{\mathbf{x}\in S}\left[\sum_{i=2}^{\infty} \frac{(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x})^T}{(i-1)^2}\, \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \in [(i-1)\theta, i\theta)\}\right]$$

4. Run the spectral tester of Proposition 26 on M_S given $\delta \leftarrow \delta$, $\lambda \leftarrow C_1\lambda^{C_1}$ and $\theta \leftarrow \frac{C_1}{2}\theta\lambda^{C_1}$, i.e., compute $\|M_S\|_{\mathrm{op}}$ and if $\|M_S\|_{\mathrm{op}} > C_1\theta\lambda^{C_1}$, then **reject**. Otherwise, **accept**.

First, suppose the test accepts. For the following, consider the vector $\mathbf{w}' \in \mathbb{R}^d$ to be an arbitrary unit vector and $\mathbf{v} \in \mathbb{R}^d$ to be the unit vector that is perpendicular to $\mathbf{w}$, lies within the plane defined by $\mathbf{w}$ and $\mathbf{w}'$ and $\langle \mathbf{v}, \mathbf{w}'\rangle \leq 0$. Then we have:

$$\begin{aligned}
&\mathbb{P}_{\mathbf{x}\sim S}[\mathrm{sign}(\langle \mathbf{w}', \mathbf{x}\rangle) \neq \mathrm{sign}(\langle \mathbf{w}, \mathbf{x}\rangle)] \leq \\
&\qquad \leq \sum_{i=1}^{\infty} \mathbb{P}_{\mathbf{x}\sim S}\Big[\underbrace{|\langle \mathbf{v}, \mathbf{x}\rangle| > \frac{\theta}{\tan\theta} \cdot (i-1)}_{\text{Implies } |\langle \mathbf{v},\mathbf{x}\rangle| > (i-1)/2} \;\&\; |\langle \mathbf{w}, \mathbf{x}\rangle| \in [(i-1)\theta, i\theta]\Big] \\
&\qquad \leq \underbrace{\mathbb{P}_{\mathbf{x}\in S}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \in [0, \theta]\right]}_{\leq C_1\theta\lambda^{C_1}} + \underbrace{4\sum_{i=2}^{\infty} \frac{\mathbb{E}_{\mathbf{x}\sim S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle| \in [(i-1)\theta, i\theta]}\right]}{(i-1)^2}}_{\langle \mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{v},\, M\ \mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{v}\rangle \leq \|M\|_{\mathrm{op}} \leq C_1\theta\lambda^{C_1}} \\
&\qquad \leq 5C_1\theta\lambda^{C_1}
\end{aligned}$$

For part (b), we suppose that the distribution $D_{\mathcal{X}}$ is indeed λ-nice. We will show that with probability at least $1 - \delta$, the tester will accept, i.e., that

$$\mathbb{P}_{\mathbf{x}\in S}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \in [0, \theta]\right] \leq C_1\theta\lambda^{C_1} \text{ and} \tag{2.3}$$

$$\|M_S\|_{\mathrm{op}} \leq C_1\theta\lambda^{C_1} \tag{2.4}$$

[6]Note that only at most $|S|$ many terms below are non-zero, hence M_S can be computed efficiently.

We first observe that the corresponding quantities under distribution $D_{\mathcal{X}}$ due to Proposition 27. In particular, we have that for some universal constant $C' > 0$

$$\mathbb{P}_{\mathbf{x}\in D_{\mathcal{X}}}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \in [0,\theta]\right] \leq C'\theta\lambda^{C'} \text{ and} \tag{2.5}$$

$$\mathbb{E}_{\mathbf{x}\in D_{\mathcal{X}}}[\langle \mathbf{v}', \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \in [c, c+\theta]\}] \leq C'\theta\lambda^{C'} \text{ for any } \mathbf{v}' \in \mathbb{S}^{d-1} \text{ and } c \geq 0 \tag{2.6}$$

If we let $M_{D_{\mathcal{X}}} = \mathbb{E}_{D_{\mathcal{X}}}[M_S]$, we get that

$$\begin{aligned}
\|M_{D_{\mathcal{X}}}\|_{\mathrm{op}} &= \sup_{\mathbf{u}\in\mathbb{S}^{d-2}} \mathbf{u}^T M_{D_{\mathcal{X}}}\mathbf{u} = \sup_{\mathbf{v}'\in\mathbb{S}^{d-1};\langle \mathbf{v}',\mathbf{w}\rangle=0} (\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{v}')^T M_{D_{\mathcal{X}}}(\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{v}') \\
&\leq \sum_{i=2}^{\infty} \frac{1}{(i-1)^2} \sup_{\mathbf{v}'\in\mathbb{S}^{d-1}} \mathbb{E}_{\mathbf{x}\in D_{\mathcal{X}}}[\langle \mathbf{v}', \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \in [c, c+\theta]\}] \\
&\leq \sum_{i=2}^{\infty} \frac{1}{(i-1)^2} \sup_{\mathbf{v}'\in\mathbb{S}^{d-1}} C'\theta\lambda^{C'} \leq \frac{C'\pi^2}{6}\theta\lambda^{C'}
\end{aligned}$$

By Proposition 26, in order to satisfy expression (2.4), it remains to show that $\mathbb{E}_{\mathbf{z}\sim D}[(\mathbf{z}_\ell \mathbf{z}_j)^2] \leq C_1\lambda^{C_1}$ for any $\ell, j \in [d]$, where $\mathbf{z}$ is defined as follows

$$\mathbf{z} = \sum_{i=2}^{\infty} \frac{\mathrm{proj}_{\perp\mathbf{w}}\,\mathbf{x}}{(i-1)} \mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle| \in [(i-1)\theta, i\theta)}.$$

Since $\mathbb{E}_{\mathbf{z}\sim D}[(\mathbf{z}_\ell \mathbf{z}_j)^2] \leq \mathbb{E}_{\mathbf{z}\sim D}[\langle \mathbf{u}, \mathbf{x}\rangle^2 \langle \mathbf{u}', \mathbf{x}\rangle^2]$, for some unit vectors $\mathbf{u}, \mathbf{u}' \in \mathbb{S}^{d-1}$ (orthogonal to $\mathbf{w}$), the desired bound follows from Proposition 27.

It remains to bound the absolute distance between the quantities of the left hand side of expressions (2.3) and (2.5). This can be achieved by an application of the Hoeffding bound, since the empirical version of the quantity is the average of independent Bernoulli random variables. □

2.6.2 Proof of Lemma 7

We restate Lemma 7 here for convenience.

Lemma 11 (Universally Testable Weak Anti-Concentration)**.** *Let D be a distribution over $\mathbb{R}^d$. Then, there is a universal constant $C > 0$ and a tester that given a unit vector $\mathbf{w} \in \mathbb{R}^d$, $\delta \in (0,1)$, $\gamma > 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{2\lambda}$ and a set S of i.i.d. samples from D with size at least $C \cdot \frac{d^4}{\sigma^2\delta}\log(d)\lambda^C$, runs in time $\mathrm{poly}(d, \lambda, \frac{1}{\sigma}, \frac{1}{\delta}, \log\left(\frac{1}{\gamma}\right))$ and satisfies the following specifications*

(*a*) *If the tester accepts S, then for any unit vector $\mathbf{v} \in \mathbb{R}^d$ with $\langle \mathbf{v}, \mathbf{w}\rangle = 0$ we have*

$$\mathbb{P}_{\mathbf{x}\in S}\left[|\langle \mathbf{v}, \mathbf{x}\rangle| \geq \frac{1}{C\lambda^C} \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] \geq \frac{1}{C\lambda^C\gamma^4}$$

(*b*) *If D is γ-Poincaré and λ-nice, then the tester accepts S with probability at least $1-\delta$.*

Proof. The testing algorithm receives a set $S \subset \mathbb{R}^d$, $\mathbf{w} \in \mathbb{S}^{d-1}, \delta \in (0,1), \gamma > 0, \lambda \geq 1$ and $\sigma \leq \frac{1}{2\lambda}$ and does the following for some sufficiently large $C_1 > 0$:

1. If $\mathbb{P}_{\mathbf{x}\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma] > 2\sigma \cdot C_1\lambda^{C_1}$, then **reject**.
2. Compute the $(d-1)\times(d-1)$ matrix M_S as follows:
$$M_S = \mathop{\mathbb{E}}_{\mathbf{x}\in S}\left[(\mathrm{proj}_{\perp \mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp \mathbf{w}}\mathbf{x})^T \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle \leq \sigma|\}\right]$$
3. Run the spectral tester of Proposition 26 on M_S given $\delta \leftarrow \delta$, $\lambda \leftarrow C_1\lambda^{C_1}$ and $\theta \leftarrow \frac{2\sigma}{C_1\lambda^{C_1}}$, i.e., **reject** if the minimum singular value of M_S is less than $\frac{2\sigma}{C_1\lambda^{C_1}}$.
4. Run the hypercontractivity tester (Prop. 24) on $S' = \{\mathrm{proj}_{\perp \mathbf{w}}\mathbf{x} : \mathbf{x} \in S \text{ and } |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}$, i.e., solve an appropriate SDP and **reject** if the solution is larger than a specified threshold. Otherwise, **accept**.

For part (a), we apply the Paley–Zygmund inequality to the random variable $Z = \langle \mathbf{v}, \mathbf{x}\rangle^2$ condtitioned on $|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma$ and get

$$\mathop{\mathbb{P}}_{\mathbf{x}\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \geq \frac{1}{2}\mathop{\mathbb{E}}_{\mathbf{x}\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] \geq \frac{(\mathbb{E}_{\mathbf{x}\in S}[\langle \mathbf{v}, \mathbf{x}\rangle^2 \mid |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma])^2}{4\,\mathbb{E}_{\mathbf{x}\in S}[\langle \mathbf{v}, \mathbf{x}\rangle^4 \mid |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma]}$$

Note that since $\langle \mathbf{v}, \mathbf{w}\rangle = 0$, we have $\langle \mathbf{v}, \mathbf{x}\rangle = \langle \mathrm{proj}_{\perp \mathbf{w}}\mathbf{v}, \mathrm{proj}_{\perp \mathbf{w}}\mathbf{x}\rangle$ (where $\|\mathbf{v}\|_2 = \|\,\mathrm{proj}_{\perp \mathbf{w}}\mathbf{v}\|_2$). Therefore, since S has passed the spectral tester as well as the tester for the probability of lying within the strip $|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma$, we have that

$$\mathop{\mathbb{E}}_{\mathbf{x}\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] = \frac{\mathbb{E}_{\mathbf{x}\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}\{|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}\right]}{\mathbb{P}_{\mathbf{x}\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma]} \geq \frac{1}{2C_1\lambda^{2C_1}}$$

Moreover, $\{\mathbf{x} \in S : |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}$ has passed the hypercontractivity tester, and therefore, according to Proposition 24 we have

$$\mathop{\mathbb{E}}_{\mathbf{x}\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^4 \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] \leq C_1 \cdot \gamma^4$$

Combining the above inequalities we conclude the proof of part (a).

For part (b), we assume that D is indeed λ-nice and γ-Poincaré. We first use Proposition 27 as well as a Hoeffding bound, to get that $\mathbb{P}_{\mathbf{x}\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma] \in [\frac{2\sigma}{C'\lambda^{C'}}, 2\sigma \cdot C'\lambda^{C'}]$ with probability at least $1-\delta/3$ over S (since $|S|$ is large enough), for some universal constant $C' > 0$. Then, we use part (ii) of Proposition 27 to lower bound the minimum eigenvalue of $M_D = \mathbb{E}_D[M_S]$ by $\frac{4\sigma}{C'\lambda^{C'}}$. Using part (iii) of Proposition 27 to bound the second moment of each of the elements of M_D, we may use Proposition 26 to get that $M_S \succeq \frac{2\sigma}{C'\lambda^{C'}} I_{d-1}$ (and our spectral test passes) with probability

at least $1-\delta/3$. It remains to show that the hypercontractivity tester will accept with probability at least $1-\delta/3$ (since, then, the result follows from a union bound).

We acquire samples from the hypercontractivity tester through rejection sampling (we keep only the samples within the strip). Since the probability of falling inside the strip is at least $\frac{2\sigma}{C'\lambda^{C'}}$, the number of samples we will keep is at least $|S'| \geq \frac{|S|\sigma}{C''\lambda^{C'}}$, for some large enough constant $C'' > 0$ (due to Chernoff bound) and with probability at least $1-\delta/6$. We now apply Lemma 9 to get that the distribution of $\mathrm{proj}_{\perp \mathbf{w}} \mathbf{x}$ conditioned on the strip $|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma$ is γ-Poincaré, since D is also γ-Poincaré. Hence, the hypercontractivity tester accepts with probability at least $1-\delta/6$ due to Proposition 24. □

2.7 Proofs from Section 2.4

2.7.1 Proof of Proposition 25

We restate Proposition 25 here for completeness.

Proposition 30 (Modification from [DKTZ20a, DKTZ20b])**.** *For a distribution $D_{\mathcal{X}\mathcal{Y}}$ over $\mathbb{R}^d \times \{\pm 1\}$ let opt be the minimum error achieved by some origin-centered halfspace and $\mathbf{w}^* \in \mathbb{S}^{d-1}$ a corresponding vector. Consider $\mathcal{L}_\sigma$ as in Equation (2.2) for $\sigma > 0$ and let $\eta < 1/2$. Let $\mathbf{w} \in \mathbb{S}^{d-1}$ with $\measuredangle(\mathbf{w}, \mathbf{w}^*) = \theta < \frac{\pi}{2}$ and $\mathbf{v} \in \mathrm{span}(\mathbf{w}, \mathbf{w}^*)$ such that $\langle \mathbf{v}, \mathbf{w}\rangle = 0$ and $\langle \mathbf{v}, \mathbf{w}^*\rangle < 0$. Then, for some universal constant $C > 0$ and any $\alpha \geq \frac{\sigma}{2\tan\theta}$ we have $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{X}\mathcal{Y}})\|_2 \geq A_1 - A_2 - A_3$, where*

$$A_1 = \frac{\alpha}{C\cdot\sigma} \cdot \mathbb{P}\left[|\langle \mathbf{v}, \mathbf{x}\rangle| \geq \alpha \;\; \textit{and} \;\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}\right]$$
$$A_2 = \frac{C}{\tan\theta} \cdot \mathbb{P}\left[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}\right] \;\; \textit{and} \;\; A_3 = \frac{C}{\sigma} \cdot \sqrt{\mathrm{opt}} \cdot \sqrt{\mathbb{E}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \cdot \mathbb{1}_{\{|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}\}}\right]}$$

Moreover, if the noise is Massart with rate η, then $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{X}\mathcal{Y}})\|_2 \geq (1-2\eta)A_1 - A_2$.

Proof. For any vector $\mathbf{x} \in \mathbb{R}^d$, let: $\mathbf{x}_\mathbf{w} = \langle \mathbf{w}, \mathbf{x}\rangle$ and $\mathbf{x}_\mathbf{v} = \langle \mathbf{v}, \mathbf{x}\rangle$. It follows that $\mathrm{proj}_V(\mathbf{x}) = \mathbf{x}_\mathbf{v}\mathbf{e}_1 + \mathbf{x}_\mathbf{w}\mathbf{e}_2$, where proj_V is the operator that orthogonally projects vectors on V. Using the fact that $\nabla_\mathbf{w}(\langle \mathbf{w}, \mathbf{x}\rangle / \|\mathbf{w}\|_2) = \mathbf{x} - \langle \mathbf{w}, \mathbf{x}\rangle \mathbf{w} = \mathbf{x} - \mathbf{x}_\mathbf{w}\mathbf{w}$ for any $\mathbf{w} \in \mathbb{S}^{d-1}$, the interchangeability of the gradient and expectation operators and the fact that ℓ'_σ is an even function we get that

$$\nabla_\mathbf{w} \mathcal{L}_\sigma(\mathbf{w}) = \mathbb{E}\Big[-\ell'_\sigma(|\langle \mathbf{w}, \mathbf{x}\rangle|) \cdot y \cdot (\mathbf{x} - \mathbf{x}_\mathbf{w}\mathbf{w})\Big]$$

Since the projection operator proj_V is a contraction, we have $\|\nabla_\mathbf{w} \mathcal{L}_\sigma(\mathbf{w})\|_2 \geq \|\mathrm{proj}_V \nabla_\mathbf{w} \mathcal{L}_\sigma(\mathbf{w})\|_2$, and we can therefore restrict our attention to a simpler, two dimensional problem. In particular,

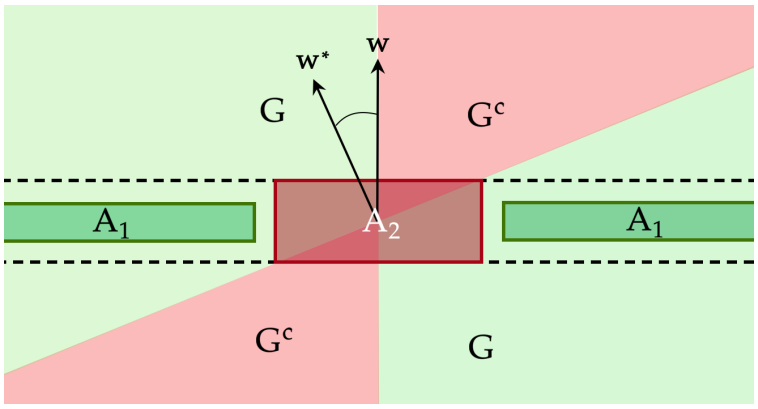

Figure 2-2: The Gaussian mass in each of the regions labelled A_1 and A_2 is proportional to the corresponding term appearing in the statement of Proposition 25.

As σ tends to 0, the Gaussian mass of region A_2 shrinks faster than the one of region A_1, since both the height (σ) and the width ($\frac{\sigma}{\tan\theta}$) of A_2 are proportional to σ, while the width of A_1 is not affected (the height is $\sigma/3$). Lemma 8 demonstrates that a similar property is universally testable under any nice Poincaré distribution.

since $\mathrm{proj}_V(\mathbf{x}) = \mathbf{x_v}\mathbf{e}_1 + \mathbf{x_w}\mathbf{e}_2$, we get

$$\begin{aligned}\|\mathrm{proj}_V \nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w})\|_2 &= \Big| \mathbb{E}\Big[-\ell'_\sigma(|\mathbf{x_w}|) \cdot y \cdot \mathbf{x_v}\Big]\Big| \\ &= \Big| \mathbb{E}\Big[-\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle) \cdot (1 - 2\,\mathbb{1}\{y \neq \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle)\}) \cdot \mathbf{x_v}\Big]\Big|\end{aligned}$$

Let $F(y, \mathbf{x})$ denote $1 - 2\,\mathbb{1}\{y \neq \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle)\}$. We may write $\mathbf{x_v}$ as $|\mathbf{x_v}| \cdot \mathrm{sign}(\mathbf{x_v})$ and let $\mathcal{G} \subseteq \mathbb{R}^2$ such that $\mathrm{sign}(\mathbf{x_v}) \cdot \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle) = -1$ iff $\mathbf{x} \in \mathcal{G}$.

Then, $\mathrm{sign}(\mathbf{x_v}) \cdot \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle) = \mathbb{1}\{\mathbf{x} \notin \mathcal{G}\} - \mathbb{1}\{\mathbf{x} \in \mathcal{G}\}$. We get

$$\begin{aligned}&\|\mathrm{proj}_V \nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w})\|_2 = \\ &\qquad = \Big| \mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|) \cdot (\mathbb{1}\{\mathbf{x} \in \mathcal{G}\} - \mathbb{1}\{\mathbf{x} \notin \mathcal{G}\}) \cdot F(y, \mathbf{x}) \cdot |\mathbf{x_v}| \cdot\Big]\Big| \\ &\qquad \geq \mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \in \mathcal{G}\} \cdot F(y, \mathbf{x}) \cdot |\mathbf{x_v}|\Big] - \mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \notin \mathcal{G}\} \cdot F(y, \mathbf{x}) \cdot |\mathbf{x_v}|\Big]\end{aligned}$$

Let $A'_1 = \mathbb{E}[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \in \mathcal{G}\} \cdot F(y, \mathbf{x}) \cdot |\mathbf{x_v}|]$ and $A'_2 = \mathbb{E}[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \notin \mathcal{G}\} \cdot F(y, \mathbf{x}) \cdot |\mathbf{x_v}|]$.

In the Massart noise case $\mathbb{E}_{y|\mathbf{x}}[F(y, \mathbf{x})] = 1 - 2\eta(\mathbf{x}) \in [1 - 2\eta, 1]$, where $1 - 2\eta > 0$. Therefore, we have that $A'_1 \geq (1 - 2\eta) \cdot \mathbb{E}[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \in \mathcal{G}\} \cdot |\mathbf{x_v}|]$. When the noise is adversarial, we have $A'_1 \geq \mathbb{E}[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \in \mathcal{G}\} \cdot |\mathbf{x_v}|] - 2\,\mathbb{E}[\ell'_\sigma(|\mathbf{x_w}|) \cdot \mathbb{1}\{\mathbf{x} \in \mathcal{G}\} \cdot \mathbb{1}\{y \neq \mathrm{sign}(\langle \mathbf{w}^*, \mathbf{x}\rangle)\} \cdot |\mathbf{x_v}|]$.

For any $\alpha \geq \frac{\sigma}{2\tan\theta}$, we have that

$$\begin{aligned}
\mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|)\cdot \mathbb{1}\{\mathbf{x}\in\mathcal{G}\}\cdot|\mathbf{x_v}|\Big] &\geq \mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|)\cdot \mathbb{1}\{\mathbf{x}\in\mathcal{G}\}\cdot \mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{6}\}}\cdot|\mathbf{x_v}|\Big] \\
&\qquad\qquad \text{(since terms are positive)} \\
&\geq \mathbb{E}\left[\frac{1}{\sigma}\cdot\mathbb{1}\{\mathbf{x}\in\mathcal{G}\}\cdot\mathbb{1}\left\{|\mathbf{x_w}|\leq\frac{\sigma}{6}\right\}\cdot|\mathbf{x_v}|\right] \quad \text{(by Proposition 28)} \\
&\geq \frac{\alpha}{\sigma}\cdot\mathbb{E}\left[\mathbb{1}\{\mathbf{x}\in\mathcal{G}\}\cdot\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{6}\}}\cdot\mathbb{1}_{\{|\mathbf{x_v}|\geq\alpha\}}\right] \\
&\geq \frac{\alpha}{\sigma}\cdot\mathbb{E}\left[\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{6}\}}\cdot\mathbb{1}_{\{|\mathbf{x_v}|\geq\alpha\}}\right] \qquad \text{(see Figure 2-2)} \\
&= \frac{\alpha}{\sigma}\cdot\mathbb{P}\left[|\mathbf{x_w}|\leq\frac{\sigma}{6}\ \text{ and }\ |\mathbf{x_v}|\geq\alpha\right] \stackrel{\text{def}}{=} A_1
\end{aligned}$$

Moreover, for some universal constant $C' > 0$, we similarly have

$$\begin{aligned}
\mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|)\cdot \mathbb{1}\{\mathbf{x}\notin\mathcal{G}\}\cdot F(y,\mathbf{x})\cdot|\mathbf{x_v}|\Big] &\leq \mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|)\cdot \mathbb{1}\{\mathbf{x}\notin\mathcal{G}\}\cdot|\mathbf{x_v}|\Big] \quad \text{(since } F(y,\mathbf{x})\leq 1\text{)} \\
&\leq \mathbb{E}\left[\frac{C'}{\sigma}\cdot\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\cdot\mathbb{1}\{\mathbf{x}\notin\mathcal{G}\}\cdot|\mathbf{x_v}|\right] \\
&\qquad\qquad \text{(by Proposition 28)} \\
&\leq \frac{C'}{\sigma}\cdot\mathbb{E}\left[\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\cdot\mathbb{1}_{\{|\mathbf{x_v}|\leq\frac{\sigma}{2\tan\theta}\}}\cdot|\mathbf{x_v}|\right] \\
&\qquad\qquad \text{(see Figure 2-2)} \\
&\leq \frac{C'}{2\cdot\tan\theta}\cdot\mathbb{E}\left[\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\cdot\mathbb{1}_{\{|\mathbf{x_v}|\leq\frac{\sigma}{2\tan\theta}\}}\right] \\
&\leq \frac{C'}{2\cdot\tan\theta}\cdot\mathbb{P}\left[|\mathbf{x_w}|\leq\frac{\sigma}{2}\right] \stackrel{\text{def}}{=} A_2
\end{aligned}$$

Hence, we have shown that, in the Massart noise case, we have $\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w})\|_2 \geq (1-2\eta)A_1 - A_2$ as desired. For the adversarial noise case, it remains to bound the following quantity

$$\begin{aligned}
2\,\mathbb{E}\Big[\ell'_\sigma(|\mathbf{x_w}|)\cdot\mathbb{1}_{\{\mathbf{x}\in\mathcal{G}\}}\cdot\mathbb{1}_{\{y\neq\text{sign}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}}\cdot|\mathbf{x_v}|\Big] \leq& \\
&\leq \frac{2C'}{\sigma}\cdot\mathbb{E}\left[\mathbb{1}_{\{\mathbf{x}\in\mathcal{G}\}}\cdot\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\cdot\mathbb{1}_{\{y\neq\text{sign}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}}\cdot|\mathbf{x_v}|\right] \\
&\leq \frac{2C'}{\sigma}\cdot\mathbb{E}\left[\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\cdot\mathbb{1}_{\{y\neq\text{sign}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}}\cdot|\mathbf{x_v}|\right] \\
&\leq \frac{2C'}{\sigma}\cdot\sqrt{\text{opt}}\cdot\sqrt{\mathbb{E}\left[|\mathbf{x_v}|^2\cdot\mathbb{1}_{\{|\mathbf{x_w}|\leq\frac{\sigma}{2}\}}\right]} \stackrel{\text{def}}{=} A_3
\end{aligned}$$

where the final inequality follows from Cauchy-Schwarz inequality. □

2.7.2 Proof of Lemma 8

We restate Lemma 8 here for convenience.

Lemma 12 (Universally Testable Structure of Surrogate Loss)**.** *Let $D_{\mathcal{XY}}$ be any distribution over $\mathbb{R}^d \times \{\pm 1\}$. Consider $\mathcal{L}_\sigma$ as in Equation* (2.2)*. Then, there is a universal constant $C > 0$ and a tester that given a unit vector $\mathbf{w} \in \mathbb{R}^d$, $\delta \in (0,1)$, $\eta < 1/2$, $\gamma > 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{C\lambda^C}$ and a set S of i.i.d. samples from $D_{\mathcal{XY}}$ with size at least $C \cdot \frac{d^4}{\sigma^2\delta}\log(d)\lambda^C$, runs in time $\mathrm{poly}(d, \lambda, \frac{1}{\sigma}, \frac{1}{\delta}, \log\left(\frac{1}{\gamma}\right))$ and satisfies the following specifications*

(*a*) *If the tester accepts S, then, the following statements are true for the minimum error opt_S achieved by some origin-centered halfspace on S and the optimum vector $\mathbf{w}_S^* \in \mathbb{S}^{d-1}$*

- *If the noise is Massart with associated rate η and $\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \leq \frac{1-2\eta}{C\lambda^C\gamma^4}$ then either $\measuredangle(\mathbf{w}, \mathbf{w}_S^*) \leq \frac{C\lambda^C(1+\gamma^4)}{1-2\eta} \cdot \sigma$ or $\measuredangle(-\mathbf{w}, \mathbf{w}_S^*) \leq \frac{C\lambda^C(1+\gamma^4)}{1-2\eta} \cdot \sigma$.*
- *If the noise is adversarial with $\mathrm{opt}_S \leq \frac{\sigma}{C\lambda^C}$ and $\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 < \frac{1}{C\lambda^C\gamma^4}$ then either $\measuredangle(\mathbf{w}, \mathbf{w}_S^*) \leq C\lambda^C(1+\gamma^4) \cdot \sigma$ or $\measuredangle(-\mathbf{w}, \mathbf{w}_S^*) \leq C\lambda^C(1+\gamma^4) \cdot \sigma$.*

(*b*) *If the marginal $D_{\mathcal{X}}$ is λ-nice and γ-Poincaré, then the tester accepts S with probability at least $1 - \delta$.*

Proof of Lemma 8. The testing algorithm receives $\mathbf{w} \in \mathbb{S}^{d-1}$, $\delta \in (0,1), \eta < 1/2$, $\gamma > 0$, $\lambda \geq 1$, $\sigma \leq \frac{1}{2\lambda}$ and a set $S \subset \mathbb{R}^d \times \{\pm 1\}$ and does the following for some sufficiently large $C_1 > 0$

1. If $\mathbb{P}_{(\mathbf{x},y)\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}] \leq \frac{\sigma}{C_1\lambda^{C_1}}$ or $\mathbb{P}_{(\mathbf{x},y)\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}] > \sigma \cdot C_1\lambda^{C_1}$, then **reject**.
2. Compute the $(d-1) \times (d-1)$ matrices M_S^+ and M_S^- as follows:
$$M_S^+ = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S}\left[(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})^T \cdot \mathbb{1}_{\{|\langle \mathbf{w},\mathbf{x}\rangle| \leq \frac{\sigma}{2}\}}\right]$$
$$M_S^- = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S}\left[(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})^T \cdot \mathbb{1}_{\{|\langle \mathbf{w},\mathbf{x}\rangle| \leq \frac{\sigma}{6}\}}\right]$$
3. Run the (maximum singular value) spectral tester of Proposition 26 on M_S^+ given $\delta \leftarrow \frac{\delta}{4}$, $\lambda \leftarrow C_1\lambda^{C_1}$ and $\theta \leftarrow \frac{C_1\sigma\lambda^{C_1}}{2}$, i.e., **reject** if the maximum singular value of M_S^+ is greater than $\sigma \cdot C_1\lambda^{C_1}$.
4. Run the (minimum singular value) spectral tester of Proposition 26 on M_S^- given $\delta \leftarrow \frac{\delta}{4}$, $\lambda \leftarrow C_1\lambda^{C_1}$ and $\theta \leftarrow \frac{2\sigma}{C_1\lambda^{C_1}}$, i.e., **reject** if the minimum singular value of M_S^- is less than $\frac{\sigma}{C_1\lambda^{C_1}}$.
5. Run the hypercontractivity tester on $S' = \{\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x} : (\mathbf{x}, y) \in S \text{ and } |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\}$, i.e., solve an appropriate SDP (see Prop. 24 with $\gamma \leftarrow \gamma$, $\delta \leftarrow \delta/4$) and **reject** if the solution is larger than a specified threshold. Otherwise, **accept**.

For part (a), we suppose that the testing algorithm has accepted S. Therefore, S has passed all the tests required for part (a) of Lemma 7 and there exists a universal constant $C' > 0$ such that

$$\mathop{\mathbb{P}}_{(\mathbf{x},y)\in S}\left[|\langle \mathbf{v}, \mathbf{x}\rangle| \geq \frac{1}{C'\lambda^{C'}} \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \sigma\right] \geq \frac{1}{C'\lambda^{C'}\gamma^4}$$

Moreover, we have $\frac{\sigma}{C'\lambda^{C'}} < \mathbb{P}_{(\mathbf{x},y)\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}] \leq \mathbb{P}_{(\mathbf{x},y)\in S}[|\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}] < \sigma \cdot C'\lambda^{C'}$ and

$$\mathop{\mathbb{E}}_{(\mathbf{x},y)\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{2}\right] \leq C'\lambda^{C'}$$
$$\mathop{\mathbb{E}}_{(\mathbf{x},y)\in S}\left[\langle \mathbf{v}, \mathbf{x}\rangle^2 \;\middle|\; |\langle \mathbf{w}, \mathbf{x}\rangle| \leq \frac{\sigma}{6}\right] \geq \frac{1}{C'\lambda^{C'}}$$

Since Proposition 25 holds for any distribution, it will also hold for the empirical distribution (uniform on S). We apply Proposition 25 with $\alpha = \frac{1}{C'\lambda^{C'}}$ to lower bound $\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2$ (or $\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(-\mathbf{w}; S)\|_2$) as follows

$$\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \geq A_1(\alpha) - A_2 - A_3 \qquad \text{(adversarial noise case)}$$
$$\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \geq (1-2\eta)\cdot A_1(\alpha) - A_2 \qquad \text{(Massart noise case)}$$

Combining the above inequalities with the bounds implied by the fact that S has passed the tests, concludes the proof of part (a), since (after observing that $\tan\theta \geq \theta$) we get

$$\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \geq \frac{3}{C\lambda^C\gamma^4} - \frac{\sqrt{C}\sigma\lambda^{C/2}}{\theta} - \sqrt{\frac{\mathrm{opt}\cdot C\cdot\lambda^C}{\sigma}} \qquad \text{(adversarial noise case)}$$
$$\|\nabla_{\mathbf{w}}\mathcal{L}_\sigma(\mathbf{w}; S)\|_2 \geq \frac{3(1-\eta)}{C\lambda^C\gamma^4} - \frac{\sqrt{C}\sigma\lambda^{C/2}}{\theta} \qquad \text{(Massart noise case)}$$

For part (b), we follow a similar recipe as the one used to prove part (b) of Lemma 7, i.e., we use the following reasoning to show that the tests will pass with probability at least $1-\delta$

1. We assume that the marginal distribution $D_{\mathcal{X}}$ is λ-nice and γ-Poincaré.
2. We use Proposition 27 to bound the values of the tested quantities under the true distribution.
3. We use appropriate concentration results (Hoeffding/Chernoff Bounds and Proposition 26) to show that, since $|S|$ is large enough, each of the empirical quantities at hand does not deviate a lot from its mean.

This concludes the proof of Lemma 8. □

2.7.3 Proof of Main Theorem

We restate the main Theorem here for convenience.

Theorem 12 (Efficient Universal Tester-Learner for Halfspaces)**.** *Let $D_{\mathcal{XY}}$ be any distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let $\mathcal{C}$ be the class of origin centered halfspaces in $\mathbb{R}^d$. Then, for any $\lambda \geq 1$, $\gamma > 0$, $\epsilon > 0$ and $\delta \in (0,1)$, there exists an universal tester-learner for $\mathcal{C}$ w.r.t. the class of λ-nice and γ-Poincaré marginals up to error $\mathrm{poly}(\lambda) \cdot (1+\gamma^4) \cdot \mathrm{opt} + \epsilon$, where $\mathrm{opt} = \min_{\mathbf{w} \in \mathbb{S}^{d-1}} \mathbb{P}_{D_{\mathcal{XY}}}[y \neq \mathit{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)]$, and error probability at most δ, using a number of samples and running time $\mathrm{poly}(d, \lambda, \gamma, \frac{1}{\epsilon}, \log \frac{1}{\delta})$.*

Moreover, if the noise is Massart with given rate $\eta < 1/2$, then the algorithm achieves error $\mathrm{opt} + \epsilon$ with time and sample complexity $\mathrm{poly}(d, \lambda, \gamma, \frac{1}{\epsilon}, \frac{1}{1-2\eta}, \log \frac{1}{\delta})$.

Proof of Theorem 11. Note that we will give the algorithm for $\delta \leftarrow \delta' = 1/3$ since we can reduce the probability of failure with repetition (repeat $O(\log \frac{1}{\delta})$ times, accept if the rate of acceptance is $\Omega(1)$ and output the halfspace achieving the minimum test error among the halfspaces returned).

For reader's convenience, we now restate the algorithm on page 77 (note that together with the algorithm we include additional detail relevant to the analysis). The algorithm receives $\lambda \geq 1$, $\gamma > 0$, $\epsilon > 0$ and $\eta \in (0, 1/2) \cup \{1\}$ (say $\eta = 1$ when we are in the agnostic case) and does the following for some appropriately large universal constant $C_1, C_2 > 0$.

1. First, create a set of parameters Σ and parameters $E = \frac{\epsilon}{C_1 \lambda^{C_1}}$ and $A > 0$ as follows. If $\eta = 1$, then Σ is an $\frac{E}{C_1 \lambda^{C_1}}$-cover of the interval $\left[0, \frac{1}{C_1 \lambda^{C_1}}\right]$ and $A = \frac{1}{C_1 \lambda^{C_1} \gamma^4}$. Otherwise, let $\Sigma = \left\{\frac{E \cdot (1-2\eta)}{C_1 \lambda^{C_1} (1+\gamma^4)}\right\}$ and $A = \frac{1-2\eta}{C_1 \lambda^{C_1} \gamma^4}$.

2. Then, draw a set S_1 of $C_2 \left(\frac{\lambda d}{\gamma \epsilon}\right)^{C_2}$ i.i.d. samples from $D_{\mathcal{XY}}$ and run PSGD, as specified in Proposition 29 with $\epsilon \leftarrow A$, $\delta \leftarrow \frac{\delta'}{C_1}$ on the loss $\mathcal{L}_\sigma$ for each $\sigma \in \Sigma$.

3. Form a list L with all the pairs of the form $(\mathbf{w}, \sigma)$ where $\mathbf{w} \in \mathbb{S}^{d-1}$ is some iterate of the PSGD subroutine performed on $\mathcal{L}_\sigma$.

4. Draw a fresh set S_2 of $C_2 \left(\frac{\lambda d}{\gamma \epsilon}\right)^{C_2}$ i.i.d. samples from $D_{\mathcal{XY}}$ and compute for each $(\mathbf{w}, \sigma) \in L$ the value $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2$. If, for some $\sigma \in \Sigma$, $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2 > A$ for all $(\mathbf{w}, \sigma) \in L$, then **reject**.

5. Update L by keeping for each $\sigma \in \Sigma$ only one pair of the form $(\mathbf{w}, \sigma)$ for which we have $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2 \leq A$.

6. Run the following tests for each $(\mathbf{w}, \sigma) \in L$ to ensure that part (a) of Lemma 8 holds for each of the elements of L, i.e., that any stationary point of the surrogate loss that lies in L is angularly close to the empirical risk minimizer (or the same holds for the inverse vector).

   - If $\mathbb{P}_{(\mathbf{x},y) \in S_2}[|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \frac{\sigma}{6}] \leq \frac{\sigma}{C_1 \lambda^{C_1}}$ or $\mathbb{P}_{(\mathbf{x},y) \in S_2}[|\langle \mathbf{w}, \mathbf{x} \rangle| \leq \frac{\sigma}{2}] > \sigma \cdot C_1 \lambda^{C_1}$, then **reject**.

- Compute the $(d-1)\times(d-1)$ matrices $M_{S_2}^+$ and $M_{S_2}^-$ as follows:

$$M_{S_2}^+ = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})^T\cdot\mathbb{1}_{\{|\langle\mathbf{w},\mathbf{x}\rangle|\le\frac{\sigma}{2}\}}\right]$$
$$M_{S_2}^- = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})^T\cdot\mathbb{1}_{\{|\langle\mathbf{w},\mathbf{x}\rangle|\le\frac{\sigma}{6}\}}\right]$$

- Run the (maximum singular value) spectral tester of Proposition 26 on $M_{S_2}^+$ given $\delta\leftarrow\frac{\delta'}{C_1}$, $\lambda\leftarrow C_1\lambda^{C_1}$ and $\theta\leftarrow\frac{C_1\sigma\lambda^{C_1}}{2}$, i.e., **reject** if the maximum singular value of $M_{S_2}^+$ is greater than $\sigma\cdot C_1\lambda^{C_1}$.
- Run the (minimum singular value) spectral tester of Proposition 26 on $M_{S_2}^-$ given $\delta\leftarrow\frac{\delta'}{C_1}$, $\lambda\leftarrow C_1\lambda^{C_1}$ and $\theta\leftarrow\frac{2\sigma}{C_1\lambda^{C_1}}$, i.e., **reject** if the minimum singular value of $M_{S_2}^-$ is less than $\frac{\sigma}{C_1\lambda^{C_1}}$.
- Run the hypercontractivity tester on $S'=\{\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x} : (\mathbf{x},y)\in S_2 \text{ and } |\langle\mathbf{w},\mathbf{x}\rangle|\le\sigma\}$, i.e., solve an appropriate SDP (see Prop. 24 with $\gamma\leftarrow\gamma$, $\delta\leftarrow\delta'/C_1$) and **reject** if the solution is larger than a specified threshold.

7. Run the following tests for each pair of the form $(\mathbf{w},\sigma)$ and $(-\mathbf{w},\sigma)$ where $(\mathbf{w},\sigma)\in L$ to ensure that part (a) of Lemma 6 is activated, i.e., that the distance of a vector from the empirical risk minimizer is an accurate proxy for the error of the corresponding halfspace. Set $\theta(\sigma)=\frac{(1+\gamma^4)\sigma}{A\gamma^4}$.

   - If $\mathbb{P}_{(\mathbf{x},y)\in S_2}[|\langle\mathbf{w},\mathbf{x}\rangle|\le\theta] > C_1\lambda^{C_1}\theta$ then **reject**.
   - Compute the $(d-1)\times(d-1)$ matrix M_{S_2} as follows[7]:

$$M_{S_2} = \mathop{\mathbb{E}}_{(\mathbf{x},y)\in S_2}\left[\sum_{i=2}^{\infty}\frac{(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})(\mathrm{proj}_{\perp\mathbf{w}}\mathbf{x})^T}{(i-1)^2}\,\mathbb{1}\{|\langle\mathbf{w},\mathbf{x}\rangle|\in[(i-1)\theta,i\theta)\}\right]$$

   - Run the spectral tester of Proposition 26 on M_S given $\delta\leftarrow\frac{\delta'}{C_1}$, $\lambda\leftarrow C_1\lambda^{C_1}$ and $\theta\leftarrow\frac{C_1}{2}\theta\lambda^{C_1}$, i.e., compute $\|M_S\|_{\mathrm{op}}$ and if $\|M_S\|_{\mathrm{op}} > C_1\theta\lambda^{C_1}$, then **reject**.

8. Otherwise, **accept** and output the vector $\mathbf{w}$ that achieves the smallest empirical error on S_2 among the vectors in the list L.

For the following, let $\alpha=1$ in the Massart noise case and $\alpha=C_1\lambda^{C_1}\gamma^4$ in the adversarial noise case. Consider also opt_{S_2} to be the error of the origin-centered halfspace with the minimum empirical error on S_2 and $\mathbf{w}_{S_2}^*$ the corresponding optimum vector.

[7]Note that only at most $|S_2|$ many terms below are non-zero, hence M_{S_2} can be computed efficiently.

Soundness. We first prove the soundness condition, i.e., that the following implication holds with probability at least $1 - \delta'$ over the samples:

$$\text{If the tester accepts, then } \mathop{\mathbb{P}}_{D_{\mathcal{XY}}}[y \neq \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)] \leq \alpha \cdot \text{opt} + \epsilon$$

The tester accepts only if for every $\sigma \in \Sigma$, we have some $\mathbf{w} \in L$ with $\|\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)\|_2 \leq A$ (step 4) and for which part (a) of each of Lemmas 8 (step 6) and 6 (step 7) is activated. Therefore, in the Massart noise case, for any $\sigma \in \Sigma$, there is some $\mathbf{w}$ such that either $(\mathbf{w}, \sigma) \in L$ or $(-\mathbf{w}, \sigma) \in L$ and also

$$\measuredangle(\mathbf{w}, \mathbf{w}^*_{S_2}) \leq \frac{1 + \gamma^4}{\gamma^4} \cdot \frac{\sigma}{A} \stackrel{\text{def}}{=} \theta \tag{2.7}$$

$$\mathop{\mathbb{P}}_{S_2}[y \neq \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)] \leq \text{opt}_{S_2} + C' \lambda^{C'} \cdot \theta \tag{2.8}$$

In the adversarial noise case, the above are true conditional on σ being such that $\text{opt}_{S_2} \leq \frac{\sigma}{C' \lambda^{C'}}$.

Therefore, in the Massart noise case, the above are true for $\sigma = \frac{E(1-2\eta)}{C_1 \lambda^{C_1}(1+\gamma^4)}$ which gives

$$\mathop{\mathbb{P}}_{S_2}[y \neq \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)] \leq \text{opt}_{S_2} + C' \lambda^{C'} E$$

In the agnostic case, condition 2.8 is true for some $\sigma \in [0, \frac{1}{C_1 \lambda^{C_1}}]$ such that

$$\frac{\sigma}{C' \lambda^{C'}} - \frac{1}{C_1 \lambda^{C_1}} \leq \text{opt}_{S_2} \leq \frac{\sigma}{C' \lambda^{C'}}$$

unless $\text{opt} > \frac{1}{C_1 C' \lambda^{C_1 + C'}}$, in which case any halfspace has error at most $1 = \text{opt} \cdot (C_1 C' \lambda^{C_1 + C'})$. Hence we get

$$\mathop{\mathbb{P}}_{S_2}[y \neq \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)] \leq \text{poly}(\lambda) \cdot (1 + \gamma^4) \cdot \text{opt}_{S_2} + C' \lambda^{C'} E$$

Soundness follows from the fact that if $|S_2|$ is sufficiently large (but still polynomial in every parameter, since the VC dimension of the class of halfspaces in $\mathbb{R}^d$ is $d + 1$), then $|\text{opt}_{S_2} - \text{opt}| \leq \frac{\epsilon}{C_1 \lambda^{C_1}(1+\gamma)^4}$ with probability at least $1 - \delta'$.

Completeness. Suppose now that the marginal is indeed λ-nice and γ-Poincaré. Then, for sufficiently large S_1, after step 3, L will contain a stationary point of $\mathcal{L}_\sigma(\,\cdot\,; D_{\mathcal{XY}})$ for each $\sigma \in \Sigma$, due to Proposition 29. If S_2 is large enough, then steps 4, 6 and 7 will each accept with probability at least $1 - \delta'/C_1$, due to part (b) of Lemmas 8 and 6, as well as the fact that each coordinate of $\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)$ has bounded second moment (Proposition 27) and therefore $\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; S_2)$ is concentrated around $\nabla_{\mathbf{w}} \mathcal{L}_\sigma(\mathbf{w}; D_{\mathcal{XY}})$ for any fixed $\mathbf{w}$ such that $(\mathbf{w}, \sigma) \in L$ (we also need a union bound over L). Hence, in total, the tester will accept with probability at least $1 - \delta'$. □

Chapter 3

Testable Learning with Distribution Shift

3.1 Chapter Overview.

Mitigating distribution shift remains one of the major challenges of machine learning. Training distributions can deviate significantly from test distributions, and pre-trained models are commonly deployed without a precise understanding of these differences. In such cases, a model may have poor performance with potentially dangerous consequences. For example, several recent studies in the AI/healthcare community highlight the lack of generalization among many AI models trained to detect disease (e.g., skin cancer or pneumonia), often due to distribution shift. As such, developing best practices for using these models in a clinical setting remains a vexing and difficult problem [ZBL$^+$18, WOD$^+$21, TCK$^+$22].

The computational landscape of traditional supervised learning— where training sets and tests are drawn from the same distribution— is by now well understood. There is a rich literature of efficient algorithms and computational hardness results for broad sets of concept classes and distributions. In contrast, little is known in terms of efficient algorithms for classification in the context of distribution shift or domain adaptation. The most common approach is to prove a generalization bound in terms of some notion of distance between $\mathcal{D}$ and $\mathcal{D}'$ [BDBCP06, BDBC$^+$10, MMR09]. These distances, however, involve an enumeration of all functions in the underlying concept class and seem difficult to compute. Other recent work requires oracles for empirical risk minimization [GKKM20, KK21] or the existence of distribution-free reliable learners, which are believed to require superpolynomial time for even simple concept classes (e.g., reliably learning conjunctions is known to be harder than PAC learning DNF formulas) [KK21, Section 4.2].

In this chapter we define a new model called *testable learning with distribution shift* (TDS learning) and show that this model does admit efficient algorithms for several well-studied concept classes and distributions. Inspired by recent work in testable learning developed in Chapter 1, Chapter 2 and [GKK23, DKK$^+$23], we allow a learner to reject unless $\mathcal{D}$ and $\mathcal{D}'$ pass an efficiently computable test. Whenever the test accepts, the learner outputs a classifier that is assured to have low error with respect to $\mathcal{D}'$. Further, we require that the test accept with high probability whenever the marginal of $\mathcal{D}$ equals the marginal of $\mathcal{D}'$. This approach allows us to take no assumptions on

$\mathcal{D}'$ whatsoever and still provide meaningful guarantees.

It is easy to see that TDS learning generalizes the traditional PAC model of learning, and, moreover, TDS learning seems considerably more challenging. For example, even an algorithm to amplify the success probability of a TDS learner is nontrivial, since we do not get to see labeled examples from $\mathcal{D}'$ (we show how to do this in Section 3.13). It is also tempting to apply property testing algorithms in this setting to "detect" when $\mathcal{D}$ is "close" to $\mathcal{D}'$, but even for simple cases, distribution testing requires an exponential (in the dimension) number of samples (see e.g. [Can22]). While testable learning and TDS learning both encounter similar issues, they are fundamentally distinct models. Specifically, the realizable setting, where there exists a classifier with zero train and test loss, is a trivial case in testable learning. We further discuss separations among these models in Section 3.1.3.

3.1.1 Our Results

Here we formally define TDS learning and summarize our main results. For readability, we have placed some notation and basic definitions in Section 3.6.

Learning Setup. Let $\mathcal{C}$ be a function class over $\mathbb{R}^d$ and D be a distribution over $\mathbb{R}^d$. Suppose $\mathcal{A}$ is given as input a set S_{train} consisting of i.i.d. examples from D labelled by some $f \in \mathcal{C}$, together with a set of i.i.d. unlabelled examples X_{test} from some distribution $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ over $\mathbb{R}^d$. The algorithm $\mathcal{A}$ is allowed to either output REJECT or (ACCEPT, $\widehat{f}$) for some concept $\widehat{f}$. The algorithm $\mathcal{A}$ is a **TDS-learning algorithm** for $\mathcal{C}$ under distribution D if it satisfies the following two properties:

1. **Soundness.** With probability $1-\delta$, if the algorithm $\mathcal{A}$ outputs (ACCEPT, $\widehat{f}$), then hypothesis $\widehat{f}$ satisfies $\mathbb{P}_{\mathbf{x}\in\mathcal{D}_{\mathcal{X}}^{\text{test}}}[f(\mathbf{x}) \neq \widehat{f}(\mathbf{x})] \leq \epsilon$.
2. **Completeness.** If $\mathcal{D}_{\mathcal{X}}^{\text{test}} = D$, then with probability $1-\delta$, the algorithm $\mathcal{A}$ accepts.

TDS Learning: the Agnostic Setting. Sometimes the training data or the testing data cannot be captured perfectly by any function in the function class $\mathcal{C}$ and, instead, follow labeled distributions $\mathcal{D}_{\mathcal{XY}}^{\text{train}}, \mathcal{D}_{\mathcal{XY}}^{\text{test}}$, where the marginal of $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$ is $\mathcal{D}_{\mathcal{X}}^{\text{train}} = D$ and $\mathcal{D}_{\mathcal{XY}}^{\text{train}}, \mathcal{D}_{\mathcal{XY}}^{\text{test}}$ are otherwise arbitrary. We extend our setup to apply in this setting as well. To this end, a key quantity is the **smallest sum** of expected training error and expected test error among all functions in the concept class $\mathcal{C}$, i.e. $\lambda = \min_{f\in\mathcal{C}} \operatorname{err}(f; \mathcal{D}_{\mathcal{XY}}^{\text{train}}) + \operatorname{err}(f; \mathcal{D}_{\mathcal{XY}}^{\text{test}})$, where $\operatorname{err}(f; \mathcal{D}_{\mathcal{XY}}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}_{\mathcal{XY}}}[y \neq f(\mathbf{x})]$. We denote this quantity as λ, and note that it is standard in the domain adaptation literature (see, e.g., [BDBCP06, BCK$^+$07, BDBC$^+$10, DLLP10]).

With this definition at hand, we modify the soundness condition to require that with probability $1-\delta$, if the algorithm $\mathcal{A}$ outputs (ACCEPT, $\widehat{f}$), then hypothesis $\widehat{f}$ satisfies $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}_{\mathcal{XY}}^{\text{test}}}[y \neq \widehat{f}(\mathbf{x})] \leq O(\lambda) + \epsilon$. In Theorem 25, we show that a dependence of $\Omega(\lambda)$ is unavoidable.

Proposition 31. *No TDS learning algorithm can have an error guarantee better than $\Omega(\lambda) + \epsilon$.*

Results. We show that TDS learning can be achieved efficiently for a number of natural high-dimension function classes. These include halfspaces, decision trees, intersections of halfspaces and low-depth formulas. See Table 3.1 for the full list.

	Function class	Training Distribution	TDS Setting	Run-time
1	Homogeneous halfspaces	Isotropic Log-Concave	Agnostic	$\mathrm{poly}\,(d/\epsilon)$ (Theorem 13)
2	General halfspaces	Standard Gaussian	Realizable	$d^{O(\log 1/\epsilon)}$ (Theorem 15)
3	General halfspaces	Standard Gaussian Uniform on $\{\pm 1\}^d$	Agnostic	$d^{\tilde{O}\left(1/\epsilon^2\right)}$ (Corollary 4)
4	Intersection of ℓ halfspaces	Standard Gaussian Uniform on $\{\pm 1\}^d$	Agnostic	$d^{\widetilde{O}(\ell^6/\epsilon^2)}$ (Corollary 4)
5	Decision trees of size s	Uniform on $\{\pm 1\}^d$	Agnostic	$d^{O(\log(s/\epsilon))}$ (Corollary 2)
6	Formulas of size s, depth ℓ	Uniform on $\{\pm 1\}^d$	Agnostic	$d^{\sqrt{s}\cdot O(\log(s/\epsilon))^{\frac{5\ell}{2}}}$ (Corollary 3)

Table 3.1: Our TDS learning results for various function classes. Since agnostic TDS learning is more general than realizable TDS learning, algorithms for the agnostic setting also apply to the realizable setting.

Given the abundance of positive results, it is natural to ask whether TDS learning can always be achieved efficiently for any function class $\mathcal{F}$ that can be efficiently PAC-learned under a distribution $\mathcal{D}$. We answer this question in the negative by proving separations between TDS learning and PAC learning. Our separations hold for the natural and well-studied function classes of monotone functions over $\{\pm 1\}^d$ and convex sets over $\mathbb{R}^d$ (under uniform distribution on $\{\pm 1\}^d$ and the standard Gaussian distribution respectively). Even though for these function classes there are well-known PAC-learning algorithms [BT96, KOS08] that run in time $2^{\tilde{O}(\sqrt{d}\,\mathrm{poly}(1/\epsilon))}$, we show that any TDS-learning algorithm for these function classes needs to run in time $2^{\Omega(d)}$.

3.1.2 Techniques

Here we summarize the technical ideas that we use to develop the TDS learning algorithms in Table 3.1.

Moment Matching/Sandwiching Polynomials. We present a general approach for obtaining TDS learning algorithms for a wide variety of function classes via a **moment matching** approach.

In brief, the algorithm for this approach is as follows:

- Estimate all the degree-k moments of $\mathcal{D}^{\text{test}}_{\mathcal{X}}$ up to a high accuracy. REJECT if some of the moments are not close to the corresponding moments of $\mathcal{D}$.
- Otherwise, fit the best degree-k polynomial p on the training data, and output (ACCEPT, $\text{sign}(p)$).

This algorithm above runs in time $d^{O(k)}$, and we show that this algorithm is a valid TDS-learning algorithm for the wide class of functions whose **$\mathcal{L}_2$-sandwiching degree** is bounded by k, which we define as follows: For an approximation parameter ϵ, the $\mathcal{L}_2$-sandwiching degree of a function f is the smallest degree for a pair of polynomials p_{down} and p_{up} satisfying: i) $p_{\text{down}}(\mathbf{x}) \leq f(\mathbf{x}) \leq p_{\text{up}}(\mathbf{x})$ for all $\mathbf{x}$ in the learning domain and ii) $\mathbb{E}_{\mathbf{x}\sim D}[(p_{\text{up}}(\mathbf{x}) - p_{\text{down}}(\mathbf{x}))^2] \leq \epsilon$.

The related notion of $\mathcal{L}_1$-sandwiching was recently used to obtain several results in testable learning [GKK23]. These results, however, do not seem to apply to TDS learning[1] More specifically, a bound on the $\mathcal{L}_1$-sandwiching degree and low-degree moment matching would imply the existence of low-degree polynomials with low test error. However, in testable learning with distribution shift, we do not have access to these polynomials, since test labels are not available. Instead, we prove a "transfer lemma" showing that we can relate the test error under $\mathcal{D}^{\text{test}}_{\mathcal{X}}$ of a polynomial to its training error under $\mathcal{D}$ by leveraging the simple fact that the squared loss between two polynomials is itself a polynomial. As such, low-degree moment matching between the training and test marginals ensures that the squared loss between any pair of low-degree polynomials is approximately preserved (Lemma 13). Absolute loss cannot be computed by a low-degree polynomial, ruling out this type of transfer lemma based on $\mathcal{L}_1$-sandwiching.

Even though we need the more stringent property of small $\mathcal{L}_2$-sandwiching degree, we show that constructions from works in the pseudorandomness literature that explicitly construct $\mathcal{L}_1$-sandwiching polynomials (e.g., [DGJ$^+$10] and [GOWZ10]) can be extended to bound the $\mathcal{L}_2$-sandwiching degree. This allows us to obtain efficient TDS learning algorithms for the classes of intersections of halfspaces, decision trees and small-depth formulas (see lines 3-6 in Table 3.1). We also note that this technique yields TDS learning algorithms not only in the realizable setting, but also in the agnostic setting.

Beyond Moment Matching. It is a natural question whether it is possible to beat the moment-matching approach. We answer this question in the affirmative by showing that for the class of halfspaces this is indeed possible. It is a standard fact that one needs polynomials of degree $\tilde{\Omega}(1/\epsilon^2)$ to ϵ-approximate halfspaces up to $\mathcal{L}_1$ error better than ϵ under the standard Gaussian distribution. Therefore the moment-matching approach requires a run-time of at least $d^{\tilde{\Omega}(1/\epsilon^2)}$ to TDS learn halfspaces under the standard Gaussian. We overcome this obstacle and give a TDS learning algorithm for halfspaces that runs in time $d^{O(\log\left(\frac{1}{\epsilon}\right))}$ (Line 2, Table 3.1).

[1] At a high level, we can only "transfer" a classifier's loss from $\mathcal{D}$ to $\mathcal{D}'$ via polynomials, and the absolute loss cannot be computed as a low-degree polynomial.

One ingredient we use to design our algorithm is what we call TDS learning via the **disagreement region** method. Suppose we are able to recover the parameters of a halfspace f^* up to some accuracy β. Then, for some points $\mathbf{x}$ in $\mathbb{R}^d$ we will know $f^*(\mathbf{x})$ with certainty, but for some others we will not. We say that the latter points form the disagreement region, and it gets smaller as β decreases. The idea is to (i) use the training data to recover the parameters of halfspace f^* up to such high accuracy β that the probability that a Gaussian sample falls into the disagreement region is very small (ii) make sure that the recovered halfspace $\widehat{f}$ generalizes on the testing dataset by checking that only a small fraction of the testing dataset falls into the disagreement region. We note that this notion of disagreement region is also widely used in active learning (see discussion in Section 3.3.1).

Although the disagreement region method gives an efficient algorithm for homogeneous (i.e. origin-centered) halfspaces (Proposition 34), it fails for general halfspaces. Indeed, in Section 3.3.2 we show that for general halfspaces under the standard Gaussian distribution the disagreement region method requires $2^{\Omega(d)}$ samples. We design a $d^{O(\log(1/\epsilon))}$-time TDS learning algorithm for general halfspaces under the Gaussian distribution by **combining** the moment matching approach with the disagreement region approach:

- Suppose the halfspace f^* is not too biased, i.e. among $d^{O(\log(1/\epsilon))}$ training samples we see labels with values of both $+1$ and -1. We show that the parameters of such a halfspace can be recovered up to a very high accuracy using only $d^{O(\log(1/\epsilon))}$ additional training samples. This allows us to leverage the disagreement region method to achieve TDS learning.
- Otherwise, the halfspace f^* is highly biased and it almost always takes the same label L on a Gaussian input. For such halfspaces there is no hope to recover their parameters with $d^{O(\log(1/\epsilon)}$ samples. Yet, we show that using the moment-matching approach with degree parameter k of only $O(\log(1/\epsilon))$ allows us to certify that even under the test distribution $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ the halfspace f^* will be biased and very likely to take the label L. Therefore, a predictor $\widehat{f}$ that assigns the label L to all points in $\mathbb{R}^d$ will generalize.

Techniques from Testable Learning. Additionally, in the setting of **agnostic** TDS learning we give an algorithm for the class of homogeneous (i.e. origin-centered) halfspaces under any isotropic log-concave distribution (see line 1 in Table 3.1). We achieve this using techniques from testable learning developed in [GKK23] and Chapter 2. The first phase of our TDS learning algorithm uses an approximate agnostic learning algorithm for halfspaces [ABL14, DKTZ20b] in order to obtain a vector $\widehat{\mathbf{v}}$, such that the homogeneous halfspace defined by $\widehat{\mathbf{v}}$ has error $O(\lambda) + \epsilon$ in the training dataset. Since the training distribution $\mathcal{D}$ is isotropic and log-concave, this means that the angle between $\widehat{\mathbf{v}}$ and the vector $\mathbf{v}$, defining the halfspace with optimal combined error on the training and testing datasets, is also at most $O(\lambda)+\epsilon$. Finally, we apply one of the core procedures from [GKK23] and Chapter 2 in order to ensure that every halfspace defined by a vector $\mathbf{v}'$ that forms an angle of at most $O(\lambda)+\epsilon$ with $\widehat{\mathbf{v}}$ agrees on at least $1-O(\lambda)-\epsilon$ fraction of the testing dataset with the halfspace defined by the vector $\widehat{\mathbf{v}}$. This allows us to certify that the halfspace defined by the vector $\widehat{\mathbf{v}}$ will indeed generalize to the testing distribution. Note that we can use tools from testable learning to remove the assumption on the training marginal; the algorithm would instead run a test

that accepts when both $\mathcal{D}_{\mathcal{X}}^{\text{train}}$ and $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ equal the target D without any assumptions on $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$ and $\mathcal{D}_{\mathcal{XY}}^{\text{test}}$ (see also Remark 3). For clarity of exposition, we postpone formal statements composing the two models to future work.

3.1.3 Related Work

Domain Adaptation. The field of *domain adaptation* has received significant attention over the past two decades (see [BDBCP06, BCK$^+$07, MMR09, BDBC$^+$10, DLLP10, RMH$^+$20] and references therein). Similar to our learning setting, domain adaptation considers scenarios where the learner has access to labeled training and unlabeled test examples and is asked to output a hypothesis with low test error without, however, being allowed to reject. [BDBCP06, BCK$^+$07, MMR09] bound the test error of an empirical risk minimizer of training data by a sum of the parameter λ and some notion of distance between the training and test marginals (discrepancy or d_A distances) which is statistically efficient to compute using unlabeled test and training examples. This implies a statistically efficient TDS learning algorithm with error $2\lambda + \epsilon$ (Section 3.11). All known algorithms for computing discrepancy distance or d_A distance, however, require exponential time even for basic classes such as halfspaces and decision trees. By allowing the learning algorithm to reject, we design computationally efficient TDS learning algorithms with error $O(\lambda) + \epsilon$ without explicitly computing the discrepancy distance.

PQ Learning. Among the learning models that capture settings with distribution shift, PQ learning (see [GKKM20] and [KK21]) is most relevant to TDS learning. In PQ learning, the learner has access to labeled training data and unlabeled test data and must output a classifier h and a set $\mathbf{X}$. The classifier needs to minimize the following two criteria simultaneously: (1) the test error of the hypothesis h on test data points that fall into the region $\mathbf{X}$ (in other words, $\mathbf{X}$ is the region where one is confident in the predictions of the hypothesis h for test data) and (2) the probability that a training example falls outside $\mathbf{X}$. [GKKM20] show that any concept class that can be agnostically learned in the distribution-free setting can be PQ learned. [KK21] improve this reduction by showing that PQ learning is equivalent to distribution-free reliable agnostic learning (see [KKM12]). The complexity of reliable learning is known to be "in between" agnostic learning and PAC learning. In particular, reliably learning conjunctions implies PAC learning DNF formulas. In Section 3.12, we show that PQ learning actually implies TDS learning.

Testable Learning. Although conceptually our definition of TDS learning is inspired by the recent line of work in testable learning developed in Chapter 1, Chapter 2 and [GKK23, DKK$^+$23], the two frameworks address very different issues. Testable learning does not address distribution shift, as it assumes that the training and testing distributions are the same distribution $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$. What the framework of testable learning does (indirectly) test is whether $\mathcal{D}_{\mathcal{X}}^{\text{train}}$ satisfies a certain assumption (e.g. Gaussianity) in order to make sure the learning algorithm gives a hypothesis $\widehat{f}$ that satisfies the agnostic learning guarantee.

As noted in Chapter 1, in the realizable setting one can trivially satisfy the definition of testable

learning by drawing a fresh set of samples and using them to validate the hypothesis $\widehat{f}$. Due to this, existing work on testable learning Chapter 1, Chapter 2 and [GKK23, DKK+23], focus on the agnostic setting, where such validation procedure cannot be applied (see Chapter 1 for further detail). In contrast to this, even in the realizable setting, no such validation procedure exists for TDS learning, as indicated by our separations between PAC learning and TDS learning for monotone functions and convex sets (see Section 3.1.1). In fact, for monotone functions and convex sets, realizable TDS learning is harder than agnostic learning as well. Furthermore, there are cases where realizable TDS learning is easier than agnostic learning (and, therefore, easier than testable agnostic learning). Here are two examples:

1. Due to statistical query lower bounds and cryptographic hardness results [GGK20, DKZ20, DKPZ21, DKR23], the run-time required to agnostically learn a halfspace under the standard Gaussian distribution is believed to be $d^{\Omega(1/\epsilon^2)}$. In contrast to this, in this chapter we show that realizable TDS learning of halfspaces with respect to the Gaussian distribution can be achieved using only $d^{O(\log 1/\epsilon))}$ run-time.

2. The agnostic learning of parity functions, even under the uniform distribution on $\{\pm 1\}^d$, is believed to require $2^{\Omega(\frac{d}{\mathrm{poly}\log d})}$ time. In strong contrast with this, the class of parity functions can be TDS-learned in the realizable setting using only $\mathrm{poly}(d/\epsilon)$ time under any distribution over $\{\pm 1\}^d$. This follows from the PQ-learning algorithm of [KK21], together with the connection between PQ learning and TDS learning (Section 3.12).

Overall, we conclude that realizable TDS learning is incomparable to regular agnostic learning. In particular, there are examples where realizable TDS learning is easier than testable agnostic learning. Moreover, realizable TDS learning is harder than PAC learning, where distributional assumptions can be verified through validation.

3.2 TDS Learning of Homogeneous Halfspaces

We provide an efficient TDS learner for the class of homogeneous halfspaces over $\mathbb{R}^d$ with respect to any given isotropic log-concave distribution that achieves error $O(\lambda) + \epsilon$, by applying results from prior work in the literature of testable learning (see Chapter 2) and agnostic learning (see [Dan15, ABL14, DKTZ20b]). We provide the following theorem and a proof sketch. The full proof can be found in Section 3.7.

Theorem 13 (Agnostic TDS learning of Halfspaces)**.** *Let $\mathcal{C}$ be the class of origin-centered halfspaces over $\mathbb{R}^d$ and $C > 0$ a sufficiently large universal constant. Let $\mathcal{A}, \mathcal{T}$ be as defined in Propositions 32 and 33. Let $m_{\mathcal{A}}$ be the sample complexity of $\mathcal{A}(\epsilon/C, \delta/4)$ and $m_{\mathcal{T}} = \frac{Cd^4}{\epsilon^2\delta}$. Then, there is an algorithm (Algorithm 1) that, given inputs $S_{\mathrm{train}}, X_{\mathrm{test}}$ of sizes $|S_{\mathrm{train}}| \geq m_{\mathcal{A}}$ and $|X_{\mathrm{test}}| \geq m_{\mathcal{T}}$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. any isotropic log-concave distribution D with error $O(\lambda) + \epsilon$ and run-time $\mathrm{poly}(d, \frac{1}{\epsilon}) \log\left(\frac{1}{\delta}\right)$, where is the accuracy parameter and δ is the failure probability.*

Leveraging training data. We first use an efficient agnostic learner on training data to recover a halfspace $\widehat{f} : \mathbf{x} \to \text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x})$ with low training error. For example, we may use a (polynomial time) algorithm by [DKTZ20b] (Proposition 33) that outputs $\widehat{f}$ with $\text{err}(\widehat{f}; \mathcal{D}^{\text{train}}_{\mathcal{XY}}) \leq O(\eta) + \epsilon$ whenever the training marginal is isotropic log-concave (η is the optimal training error). There are other similar results in the literature of agnostic learning (e.g., see [ABL14]), but we use [DKTZ20b] as it is more convenient for our setting.

Approximate parameter recovery. Let $\mathbf{v}^*$ be the parameter vector corresponding to the halfspace f^* that minimizes the common train and test error, i.e., $\text{err}(f^*; \mathcal{D}^{\text{train}}_{\mathcal{XY}}) + \text{err}(f^*; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) = \lambda$. Then, we have $\mathbb{P}_{\mathcal{D}^{\text{train}}_{\mathcal{X}}}[\text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \text{sign}(\mathbf{v}^* \cdot \mathbf{x})] \leq \text{err}(\widehat{f}; \mathcal{D}^{\text{train}}_{\mathcal{XY}}) + \text{err}(f^*; \mathcal{D}^{\text{train}}_{\mathcal{XY}}) \leq O(\eta) + \epsilon + \lambda = O(\lambda) + \epsilon$. Since $\mathcal{D}^{\text{train}}_{\mathcal{X}} = D$ is isotropic log-concave, it is known that the disagreement over $\mathcal{D}^{\text{train}}_{\mathcal{X}}$ between two halfspaces is proportional to the angular distance between their parameters, i.e., $\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*) = O(\mathbb{P}_{\mathcal{D}^{\text{train}}_{\mathcal{X}}}[\text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \text{sign}(\mathbf{v}^* \cdot \mathbf{x})])$, which we have bounded by $O(\lambda + \epsilon)$.

Testing phase. We have shown that $\widehat{\mathbf{v}}$ is geometrically close to $\mathbf{v}^*$, which achieves test error at most λ, by definition. It remains to certify that the test marginal behaves like an isotropic log-concave distribution with respect to $\widehat{\mathbf{v}}$, i.e., for a large enough set of i.i.d. examples X_{test} from $\mathcal{D}^{\text{test}}_{\mathcal{X}}$ and for any $\mathbf{v}' \in \mathbb{S}^{d-1}$ we have that $\frac{1}{|X_{\text{test}}|} \sum_{\mathbf{x} \in X_{\text{test}}} \mathbb{1}\{\text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \text{sign}(\mathbf{v}' \cdot \mathbf{x})\} := \mathbb{P}_{X_{\text{test}}}[\text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \text{sign}(\mathbf{v}' \cdot \mathbf{x})] = O(\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}'))$, because then we will be able to bound the empirical test error of $\widehat{f}$ by $\lambda + O(\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*))$, which is $O(\lambda + \epsilon)$. The result then would follow by standard VC dimension arguments.

It turns out that Chapter 2 on testable learning has provided an efficient tester that achieves exactly what we need. Note that the proof of the following proposition (Lemma 6 in Chapter 2) is nontrivial, requiring estimation of low-order moments and careful conditioning. We can apply this to our setting, because it only requires access to the marginal distribution.

Proposition 32 (Tester of Local Halfspace Disagreement, Consequence of Lemma 6 in Chapter 2)**.** *Let D be a distribution over $\mathbb{R}^d$, $\mathbf{v}_1 \in \mathbb{S}^{d-1}$, $\theta \in (0, \pi/4]$, $\delta \in (0, 1)$ and $C > 0$ a sufficiently large universal constant. Then, there is an algorithm $\mathcal{T}(\theta, \delta)$ that, upon drawing at least $\frac{Cd^4}{\theta^2 \delta}$ examples X from D and in time $\text{poly}(d, \frac{1}{\theta}, \frac{1}{\delta})$ either accepts or rejects and satisfies the following.*

(*a*) *If $\mathcal{T}$ accepts, then for any $\mathbf{v}_2 \in \mathbb{R}^d$ with $\measuredangle(\mathbf{v}_1, \mathbf{v}_2) \leq \theta$, it holds*

$$\mathbb{P}_{\mathbf{x} \sim X}[\textit{sign}(\mathbf{v}_1 \cdot \mathbf{x}) \neq \textit{sign}(\mathbf{v}_2 \cdot \mathbf{x})] \leq C \measuredangle(\mathbf{v}_1, \mathbf{v}_2)$$

(*b*) *If D is isotropic log-concave, then $\mathcal{T}$ accepts with probability at least $1 - \delta$.*

3.3 TDS Learners for General Halfspaces

3.3.1 Warm-Up: Disagreement-Based TDS Learning

We provide a general TDS learner for the realizable setting, based on the notion of disagreement regions from active learning. Not only is this approach interesting in and of itself, but it will also be useful in Section 3.3.2 where we present our main result for TDS learning of general halfspaces in the realizable setting. The main idea is to testably bound the probability that a test example falls in some region $\mathbf{D}$, whose mass with respect to the target distribution becomes smaller as the number of training samples increases and, also, the output of the training algorithm achieves low error on any distribution that assigns small mass to $\mathbf{D}$. It turns out that the quantity $\mathbb{P}_{\mathbf{x}\sim D}[\mathbf{x} \in \mathbf{D}]$, where D is some given distribution over a space $\mathcal{X} \subseteq \mathbb{R}^d$, is a well-studied notion in the literature of active learning (see [CAL94, Han09, BBL06, Han11, Han14, BHV10, Han07] and references therein). We now provide a formal definition for the disagreement region.

Definition 10 (Disagreement Region)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$, D a distribution over $\mathcal{X}$ and $\mathcal{C}$ a concept class of functions that map $\mathcal{X}$ to $\{\pm 1\}$. For $\epsilon > 0$ and $f \in \mathcal{C}$, we define the ϵ-disagreement region of f under D, $\mathbf{D}_\epsilon(f; D)$ as the subset of $\mathcal{X}$ such that if $\mathbf{x} \in \mathbf{D}_\epsilon(f; D)$, then there are $f_1, f_2 \in \mathcal{C}$ with $\mathrm{err}(f_1, f; D) \leq \epsilon$, and $\mathrm{err}(f_2, f; D) \leq \epsilon$ and $f_1(\mathbf{x}) \neq f_2(\mathbf{x})$.*

In the literature of active learning, the quantity of interest is called the disagreement coefficient and is defined for a concept class $\mathcal{C}$ and a distribution D as follows (see, e.g., [Han14]).

$$\theta(\epsilon) = \sup_{f\in\mathcal{C}} \sup_{\epsilon'>\epsilon} \frac{\mathbb{P}_{\mathbf{x}\sim D}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(f; D)]}{\epsilon'} \tag{3.1}$$

In particular, for active learning, is is crucial that $\theta(\epsilon)$ is asymptotically bounded by a slowly increasing function of $1/\epsilon$ (e.g., $O(\log(1/\epsilon))$), since bounds on the disagreement coefficient directly provide rates on the label complexity of disagreement-based active learning, up to logarithmic factors [Han11]. In our setting, meaningful results are obtained even when $\theta(\epsilon) = O(1/\epsilon^{1-c})$ for any constant $c \in (0, 1)$. Moreover, we also focus on the dependence of the disagreement coefficient on other relevant parameters, like the dimension d. To emphasize this, in what follows, we will use the notation $\theta(\epsilon, d)$ to refer to the disagreement coefficient. We obtain the following result, which implies, for example, a polynomial improvement in the sample complexity bound of realizable TDS learning of homogeneous halfspaces w.r.t. the Gaussian compared to the TDS learner we proposed in Theorem 13 for the agnostic setting (see also Section 3.8.1).

Theorem 14 (Disagreement-Based TDS learning)**.** *Let $\mathcal{C}$ be the class of concepts that map $\mathcal{X} \subseteq \mathbb{R}^d$ to $\{\pm 1\}$ with VC dimension $\mathrm{VC}(\mathcal{C})$, let D a distribution over $\mathcal{X}$ and $C > 0$ a sufficiently large universal constant. Suppose that we have access to an ERM oracle for PAC learning $\mathcal{C}$ under D and membership access to $\mathbf{D}_{\epsilon'}(f; D)$ for any given $f \in \mathcal{C}$ and $\epsilon' > 0$. Then, there is an algorithm (Algorithm 3) that given inputs of sizes $|S_{\mathrm{train}}| \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon'} \log\left(\frac{1}{\epsilon'\delta}\right)$ and $|X_{\mathrm{test}}| \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon^2} \log\left(\frac{1}{\epsilon\delta}\right)$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. D that calls the ϵ'-ERM oracle once and the ϵ'-membership oracle $|S_{\mathrm{train}}|$ times, where ϵ is the accuracy parameter, δ is the failure probability and ϵ' such that $\epsilon' \cdot \theta(\epsilon', d) \leq \epsilon/2$.*

3.3.2 Beyond Disagreement: TDS Learners for General Halfspaces

We give a TDS-learning algorithm for the class of halfspaces under the standard Gaussian distribution. The algorithm runs in quasi-polynomial time in all relevant parameters and, contrary to the case of homogeneous halfspaces, works in a setting where efficient parameter recovery is not possible. This happens because when a general halfspace has arbitrarily large bias, it is possible, for example, that all of the training examples have the same label.

In particular, applying a pure disagreement-based TDS learning framework (Theorem 14) in the case of *general halfspaces* can only give exponential-time algorithms for this problem. To illustrate this, imagine that the ground truth is a general halfspace with bias $\tau = \sqrt{d}$ but unknown direction $\mathbf{v} \in \mathbb{S}^{d-1}$. Then, any general halfspace $\mathbf{x} \mapsto \text{sign}(\mathbf{v}' \cdot \mathbf{x} - \tau)$ with the same bias is $\exp(-\Omega(d))$-close to the ground truth with respect to the Gaussian distribution, due to standard Gaussian concentration, i.e., $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\text{sign}(\mathbf{v} \cdot \mathbf{x} - \tau) \neq \text{sign}(\mathbf{v}' \cdot \mathbf{x} - \tau)] \leq \mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\text{sign}(\mathbf{v} \cdot \mathbf{x} - \tau) \neq \text{sign}(-\mathbf{v} \cdot \mathbf{x} - \tau)]$, which is upper bounded by $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[|\mathbf{v} \cdot \mathbf{x}| > \sqrt{d}] \leq 2\exp(-d/2)$. Let $\epsilon' = 2\exp(-d/2)$. Suppose that ERM returns a halfspace $\widehat{f}$ that is ϵ'-close to the ground truth but has bias τ. Any $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\|_2 \geq \sqrt{d}$, falls within the disagreement region $\mathbf{D}_{\epsilon'}(\widehat{f}; \mathcal{N}(0, I_d))$ and therefore $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(\widehat{f}; \mathcal{N}(0, I_d))]$ is constant. This implies that running the ERM oracle on training data even up to exponentially small accuracy $\epsilon' = \exp(-\Omega(d))$ does not meet the requirement of Theorem 14 (see also [EYW12]) that the disagreement coefficient is bounded as $\epsilon' \cdot \theta(\epsilon', d) \leq \epsilon/2$.

In order to overcome this obstacle, we perform a case analysis that depends on the bias of the unknown halfspace. If the bias is bounded, then we may use a disagreement-based approach, since we can approximately recover the true parameters of the unknown halfspace using training data and it suffices to verify that the test distribution does not amplify the error between any pair of halfspaces close to the obtained approximations of the true parameters. Now, consider the case when the bias is large. We may assume without loss of generality the constant hypothesis $+1$ has low training error (since the ground truth has large bias and the marginal is Gaussian). If we can certify that the test marginal is sufficiently concentrated in every direction, then this hypothesis must also have small test error. To certify concentration for the test distribution's marginals, we use a moment-matching approach. Checking the moment matching condition only up to degree $O(\log(\epsilon))$ turns out to be sufficient to certify the type of concentration we need. We thus obtain a quasi-polynomial TDS learning algorithm for general halfspaces with respect to the Gaussian distribution. Since the probability of success can be amplified through repetition (see Proposition 44), we provide a result with constant failure probability. For the full proof, see Section 3.8.2.

Theorem 15 (TDS learning of General Halfspaces)**.** *Let $\mathcal{C}$ be the class of general halfspaces over $\mathbb{R}^d$ and $C > 0$ a sufficiently large universal constant. Then, there is an algorithm (Algorithm 4) that, given inputs of size $|S_{\text{train}}| = |X_{\text{test}}| = Cd^{C\log 1/\epsilon}$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. $\mathcal{N}(0, I_d)$ with run-time $d^{O(\log 1/\epsilon)}$, where ϵ is the accuracy parameter, and the failure probability δ is at most 0.01.*

Compared to Theorem 14, our approach here incurs an increase in the amount of test samples required (from $\text{poly}(d, 1/\epsilon)$ to $d^{O(\log(1/\epsilon))}$, used for moment matching) but significantly decreases

the amount of training samples required (from $\exp(\Omega(d))$ to $d^{O(\log(1/\epsilon))}$).

3.4 TDS Learning through Moment Matching

In the previous section, we provided a TDS learner for general halfspaces in the realizable setting that requires ideas beyond parameter recovery and testably bounding the probability of falling in the disagreement region. Crucially, Theorem 15 uses a moment-matching approach in the case when the bias of the unknown halfspaces is large. As is explained in this section, we show that the moment-matching approach can actually provide a generic result which demonstrates that $\mathcal{L}_2$-sandwiching (see Definition 11) implies TDS learning, even in the non-realizable setting. We also instantiate our framework to several important concept classes (halfspace intersections, decision trees and Boolean formulas) with respect to the Gaussian and uniform distributions, by applying constructions from pseudorandomness literature to bound the $\mathcal{L}_2$-sandwiching degree of each of these classes and acquire entries 3-6 in Table 3.1.

We provide a general theorem, which demonstrates that $\mathcal{L}_2$-sandwiching implies TDS learning under some additional natural assumptions about the target marginal distribution, which are satisfied by the standard Gaussian distribution over $\mathbb{R}^d$ and the uniform distribution on $\{\pm 1\}^d$. While it is known that $\mathcal{L}_1$-sandwiching implies testable learning (see [GKK23]), we require the stronger notion of $\mathcal{L}_2$-sandwiching. In particular, while $\mathcal{L}_1$-sandwiching would (testably) imply the existence of low degree polynomials with low test error, we do not get to see labeled examples from $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$. Moreover, we cannot a priori assume that the output of the training algorithm is a sandwiching polynomial, even if we know one exists.

In our analysis, we crucially use the fact that the square of the difference between two polynomials is itself a polynomial whose coefficients and degree are bounded by the degree and coefficient bounds of the original polynomials. Crucially, this enables us to use the following transfer lemma which relates the squared distance between polynomials under the test distribution to their squared distance under the training distribution. In what follows, we use the notation $\mathbf{x}^\alpha = \prod_{i\in[d]} \mathbf{x}_i^{\alpha_i}$, where $\alpha \in \mathbb{N}^d$.

Lemma 13 (Informal, Transfer Lemma for Square Loss, see Lemma 16)**.** *Let D be a distribution over $\mathcal{X} \subseteq \mathbb{R}^d$ and X_{test} a (multi)set of points in $\mathbb{R}^d$. If $\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}^\alpha] \approx \mathbb{E}_{\mathbf{x}\sim D}[\mathbf{x}^\alpha]$ for all $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \le 2k$, then for any degree k polynomials p_1, p_2 with bounded coefficients, it holds*

$$\frac{1}{|X_{\text{test}}|}\sum_{\mathbf{x}\in X_{\text{test}}} (p_1(\mathbf{x}) - p_2(\mathbf{x}))^2 \approx \mathbb{E}_{\mathbf{x}\sim D}[(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2]$$

Moreover, we use the fact that, due to the $\mathcal{L}_2$-sandwiching assumption, we can bound quantities of the form $\mathbb{E}[(p(\mathbf{x}) - f(\mathbf{x}))^2]$ for $f \in \mathcal{C}$ from above by $O(\mathbb{E}[(p(\mathbf{x}) - p_{\text{down}}(\mathbf{x}))^2] + \mathbb{E}[(p_{\text{down}}(\mathbf{x}) - p_{\text{up}}(\mathbf{x}))^2])$, irrespective of the distribution that the expectations are taken over. Over the training distribution, the quantity $\mathbb{E}_D[(p_{\text{down}}(\mathbf{x}) - p_{\text{up}}(\mathbf{x}))^2]$ is small via the definition of $\mathcal{L}_2$-sandwiching degree, and the quantity $\mathbb{E}_D[(p(\mathbf{x}) - f(\mathbf{x}))^2]$ because p is obtained from $\mathcal{L}_2$ polynomial regression. If $p, p_{\text{down}}, p_{\text{up}}$ are all low degree and the dataset X_{test} matches low-degree moments with D, then

we may apply Lemma 13 to bound $\frac{1}{|X_{\text{test}}|}\sum_{\mathbf{x}\in X_{\text{test}}}[(p(\mathbf{x})-f(\mathbf{x}))^2]$. Once it is shown that p fits f well on the testing dataset X_{test}, standard generalization bounds allows us to conclude that it will also predict f well on the testing distribution. Therefore, by running polynomial regression on training data to obtain p and testing whether the empirical test moments match the moments of the training distribution, we acquire the following result, whose proof can be found in Section 3.9.

Theorem 16 ($\mathcal{L}_2$-sandwiching implies TDS Learning)**.** *Let D be a distribution over a set $\mathcal{X}\subseteq\mathbb{R}^d$ and let $\mathcal{C}\subseteq\{\mathcal{X}\to\{\pm1\}\}$ be a concept class. Let $\epsilon,\delta\in(0,1)$, $\epsilon'=\epsilon/100$ $\delta'=\delta/2$ and assume that the following are true.*

(*i*) *($\mathcal{L}_2$-Sandwiching) The ϵ'-approximate $\mathcal{L}_2$ sandwiching degree of $\mathcal{C}$ under D is at most k with coefficient bound B.*

(*ii*) *(Moment Concentration) If $X\sim D^{\otimes m}$ and $m\geq m_{\text{conc}}$ then, with probability at least $1-\delta'$, we have that for any $\alpha\in\mathbb{N}^d$ with $\|\alpha\|_1\leq k$ it holds $|\mathbb{E}_D[\mathbf{x}^\alpha]-\frac{1}{|X|}\sum_{\mathbf{x}\in X}\mathbf{x}^\alpha|\leq\frac{\epsilon'}{B^2d^{4k}}$.*

(*iii*) *(Generalization) If $S\sim\mathcal{D}_{\mathcal{XY}}^{\otimes m}$ where $\mathcal{D}_{\mathcal{XY}}$ is any distribution over $\mathcal{X}\times\{\pm1\}$ such that $\mathcal{D}_{\mathcal{X}}=D$ and $m\geq m_{\text{gen}}$ then, with probability at least $1-\delta'$, for any degree-k polynomial p with coefficient bound B it holds $|\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2]-\frac{1}{|S|}\sum_{(\mathbf{x},y)\in S}[(y-p(\mathbf{x}))^2]|\leq\epsilon'$.*

Then, there is an algorithm (Algorithm 5) that, upon receiving $m_{\text{train}}\geq m_{\text{gen}}$ labelled samples S_{train} from the training distribution and $m_{\text{test}}\geq C\cdot\frac{d^k+\log(1/\delta)}{\epsilon^2}+m_{\text{conc}}$ unlabelled samples X_{test} from the test distribution (where $C>0$ is a sufficiently large universal constant), runs in time $\text{poly}(|S_{\text{train}}|,|X_{\text{test}}|,d^k)$ and TDS learns $\mathcal{C}$ with respect to D up to error $32\lambda+\epsilon$ and with failure probability δ.

3.5 Lower Bounds for Monotone Functions and Convex Sets in Realizable Setting

We provide three lower bounds for TDS learning. The first one shows that TDS learning the class of monotone functions over $\{\pm1\}^d$ with respect to the uniform distribution requires an exponential number of examples from either the training or the test distribution, which implies a separation with regular agnostic learning. The second lower bound shows that TDS learning the class of indicators of convex sets also requires an exponential in the dimension number of samples. The third lower bound demonstrates that a linear dependence on the error term λ (as defined in Equation (3.2)) is necessary for TDS learning in the non-realizable setting and can be found in Section 3.10.3.

Recent work on testable learning (which is a generalization of the classical agnostic learning framework, see Chapter 1 and [GKK23]) has demonstrated that the class of monotone functions over $\{\pm1\}^d$ cannot be testably learned with respect to the uniform distribution unless the learner draws at least $2^{\Omega(d)}$ training samples. Since the class of monotone functions can be agnostically learned in time $2^{\tilde{O}(\sqrt{d})}$ with respect to the uniform distribution over the hypercube $\{\pm1\}^d$, this

implies that testable (agnostic) learning is strictly harder than regular agnostic learning. We show that the lower bound of $2^{\Omega(d)}$ extends to the problem of TDS learning monotone functions even in the realizable setting. Recall that we have shown that we can TDS learn halfspaces with respect to the standard Gaussian distribution in the realizable setting in time $d^{O(\log(1/\epsilon))}$ (Theorem 15) but it is known that, for agnostic learning, any SQ algrorithm for the problem requires time $d^{\Omega(1/\epsilon^2)}$ (see [GGK20, DKZ20, DKPZ21]). Therefore, we conclude that realizable TDS learning and agnostic learning are incomparable. We now provide our lower bound. For the proof, see Section 3.10.

Theorem 17 (Hardness of TDS Learning Monotone Functions)**.** *Let the accuracy parameter ϵ be at most 0.1 and the success probability parameter δ also be at most 0.1. Then, in the realizable setting, any TDS learning algorithm for the class of monotone functions over $\{\pm 1\}^d$ with accuracy parameter requires either $2^{0.04d}$ training samples or $2^{0.04d}$ testing samples for all sufficiently large values of d.*

We now provide a lower bound for convex sets (see also Section 3.10). Since the class of indicators of convex sets can be agnostically learned in time $2^{\tilde{O}(\sqrt{d})}$ with respect to the Standard Gaussian on $\mathbb{R}^d$, the following theorem implies yet another separation between agnostic learning and realizable TDS learning in the distribution specific setting under the Gaussian distribution for a well-studied concept class.

Theorem 18 (Hardness of TDS Learning Convex Sets)**.** *Let the accuracy parameter ϵ be at most 0.1 and the success probability parameter δ also be at most 0.1. Then, in the realizable setting, any TDS learning algorithm for the class of indicators of convex sets under the standard Gaussian distribution on $\mathbb{R}^d$ requires either $2^{0.04d}$ training samples or $2^{0.04d}$ testing samples for all sufficiently large values of d.*

Remark 1. *In Proposition 43 of the Appendix, we show that TDS learning is not harder than PQ learning (which is a related learning primitive, see [GKKM20, KK21]). [KK21] show that the class of parities over $\{\pm 1\}^d$ can be efficiently PQ learned, which provides another example where TDS learning is easier than agnostic learning.*

3.6 Notation and Basic Definitions

We let $\mathcal{X} \subseteq \mathbb{R}^d$ and, in particular, $\mathcal{X}$ will either be the d-dimensional hypercube $\{\pm 1\}^d$ or the d-dimensional Euclidean space $\mathbb{R}^d$. For a distribution D over $\mathcal{X}$, we use $\mathbb{E}_D$ (or $\mathbb{E}_{\mathbf{x}\sim D}$) to refer to the expectation over distribution D and for a given (multi)set X, we use $\mathbb{E}_X$ (or $\mathbb{E}_{\mathbf{x}\sim X}$) to refer to the expectation over the uniform distribution on X (i.e., $\mathbb{E}_{\mathbf{x}\sim X}[g(\mathbf{x})] = \frac{1}{|X|}\sum_{\mathbf{x}\in X} g(\mathbf{x})$, counting possible duplicates separately). We let $\mathbb{R}_+ = (0, \infty)$.

For a function $p : \mathcal{X} \to \mathbb{R}$ and $r \in \mathbb{N}$, we define the $\mathcal{L}_r$ norm of p under D as $\|p\|_{\mathcal{L}_r(D)} = \mathbb{E}_{\mathbf{x}\sim D}[p(\mathbf{x})^r]^{\frac{1}{r}}$. For $\mathbf{x} \in \mathcal{X}$ where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$ and for $\alpha \in \mathbb{N}^d$, we denote with $\mathbf{x}^\alpha$ the product $\prod_{i\in[d]} \mathbf{x}_i^{\alpha_i}$, $\mathrm{M}_\alpha = \mathbb{E}[\mathbf{x}^\alpha]$ and $\|\alpha\|_1 = \sum_{i\in[d]} \alpha_i$. For a polynomial p over $\mathbb{R}^d$ and $\alpha \in \mathbb{N}^d$, we denote with p_α the coefficient of p corresponding to $\mathbf{x}^\alpha$, i.e., we have $p(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^d} p_\alpha \mathbf{x}^\alpha$. If p

is a polynomial over $\{\pm 1\}^d$, then we can always express it in a unique multilinear form, so we will only use coefficients p_α with $\alpha \in \{0,1\}^d$, i.e., $p(\mathbf{x}) = \sum_{\alpha \in \{0,1\}^d} p_\alpha \mathbf{x}^\alpha$. We define the degree of p and denote $\deg(p)$ the maximum degree of a monomial whose coefficient in p is non-zero, i.e., $\deg(p) = \max\{\|\alpha\|_1 : p_\alpha \neq 0\}$.

We denote with $\mathbb{S}^{d-1}$ the $d-1$ dimensional sphere on $\mathbb{R}^d$. For any $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$, we denote with $\mathbf{v}_1 \cdot \mathbf{v}_2$ the inner product between $\mathbf{v}_1$ and $\mathbf{v}_2$ and we let $\measuredangle(\mathbf{v}_1, \mathbf{v}_2)$ be the angle between the two vectors, i.e., the quantity $\theta \in [0, \pi]$ such that $\|\mathbf{v}_1\|_2 \|\mathbf{v}_2\|_2 \cos(\theta) = \mathbf{v}_1 \cdot \mathbf{v}_2$. For $\mathbf{v} \in \mathbb{R}^d, \tau \in \mathbb{R}$, we call a function of the form $\mathbf{x} \mapsto \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})$ an origin-centered (or homogeneous) halfspace and a function of the form $\mathbf{x} \mapsto \mathrm{sign}(\mathbf{v} \cdot \mathbf{x} - \tau)$ a general halfspace over $\mathbb{R}^d$.

$\mathcal{L}_2$-sandwiching degree. We now define the notion of $\mathcal{L}_2$-sandwiching polynomials for a function f with respect to a distribution D, i.e., a pair of polynomials such that one of them is pointwise above f, the other one is pointwise below f and the $\mathcal{L}_2$ distance between the two polynomials with respect to D is small. While the notion of L_1 sandwiching polynomials is standard in the literature of pseudorandomness (see, e.g., [Baz09]) and has applications to testable learning (see Chapter 2), in order to obtain our main results, we make use of the stronger notion of $\mathcal{L}_2$-sandwiching polynomials which we define below.

Definition 11 ($\mathcal{L}_2$-sandwiching polynomials)**.** *Consider a product set $\mathcal{X}$ and a distribution D over $\mathcal{X}$. For $\epsilon > 0$ and $f : \mathcal{X} \to \{\pm 1\}$, we say that the polynomials $p_{\mathrm{up}}, p_{\mathrm{down}} : \mathcal{X} \to \mathbb{R}$ are ϵ-approximate $\mathcal{L}_2$-sandwiching polynomials for f under D if the following are true.*

1. *$p_{\mathrm{down}}(\mathbf{x}) \leq f(\mathbf{x}) \leq p_{\mathrm{up}}(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$.*
2. *$\|p_{\mathrm{up}} - p_{\mathrm{down}}\|^2_{\mathcal{L}_2(D)} \leq \epsilon$*

Moreover, for $\epsilon > 0$, a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}\}$ and $k, B > 0$, we say that the ϵ-approximate $\mathcal{L}_2$-sandwiching degree of $\mathcal{C}$ under D is at most k and with coefficient bound B if for any $f \in \mathcal{C}$ there are ϵ-approximate $\mathcal{L}_2$-sandwiching polynomials $p_{\mathrm{up}}, p_{\mathrm{down}}$ for f such that $\deg(p_{\mathrm{up}}), \deg(p_{\mathrm{down}}) \leq k$ and each of the coefficients of $p_{\mathrm{up}}, p_{\mathrm{down}}$ are absolutely bounded by B.

Learning Setup. Consider $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ to be distributions over $\mathcal{X} \times \{\pm 1\}$ and let $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ be the corresponding marginal distributions on $\mathcal{X} \subseteq \mathbb{R}^d$. Our tester-learners receive labelled examples from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ and unlabelled examples from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ and their goal is to produce a hypothesis with low error on $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ or potentially reject but only if distribution shift is detected. Given a hypothesis class $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}\}$, $h_1, h_2 : \mathcal{X} \to \{\pm 1\}$ and distributions $\mathcal{D}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ over $\mathcal{X} \times \{\pm 1\}$, we define $\mathrm{err}(h_1; \mathcal{D}_{\mathcal{XY}}) = \mathbb{P}_{(\mathbf{x},y) \sim \mathcal{D}_{\mathcal{XY}}}[y \neq h_1(\mathbf{x})]$ and $\mathrm{err}(h_1, h_2; \mathcal{D}_{\mathcal{X}}) = \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{X}}}[h_1(\mathbf{x}) \neq h_2(\mathbf{x})]$ as well as the following quantity, which is standard in the domain adaptation literature (see, e.g., [BDBCP06, BCK+07, BDBC+10, DLLP10]).

$$\lambda(\mathcal{C}; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) := \min_{f \in \mathcal{C}} \{\mathrm{err}(f; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) + \mathrm{err}(f; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}})\}, \text{ attained by } f^* \in \mathcal{C} \tag{3.2}$$

Observe that parameter λ becomes small whenever the training and test errors can be simultaneously minimized by a common classifier in $\mathcal{C}$. Clearly, if there is no relationship between the training and test distributions, then using data from the training distribution does not reveal any information about the test distribution and, therefore, learning is hopeless (see also Theorem 25). We will assume (as is common in the domain adaptation literature) that the parameter λ is a valid choice for quantifying the relationship between the training and test distributions, in the sense that considering λ to be small is not unrealistic. In particular, we will partly focus on the following setting where λ is zero. To distinguish between the two settings, we say that we are in the **agnostic setting** when $\lambda \geq 0$ (arbitrary) and in the **realizable setting** when $\lambda = 0$. When $\lambda = 0$, there exists a classifier in $\mathcal{C}$ that achieves both zero training loss and test loss and we therefore refer to this setting as realizable. Another (slightly more specific) way to view the realizable setting is by considering the labelled distribution $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ (resp. $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$) formed as follows: for some $f^* \in \mathcal{C}$, draw an example $\mathbf{x}$ from $\mathcal{D}^{\text{train}}_{\mathcal{X}}$ (resp. $\mathcal{D}^{\text{test}}_{\mathcal{X}}$) and form the pair $(\mathbf{x}, y) \sim \mathcal{D}^{\text{train}}_{\mathcal{XY}}$ (resp. $(\mathbf{x}, y) \sim \mathcal{D}^{\text{test}}_{\mathcal{XY}}$) by setting $y = f^*(\mathbf{x})$. We now provide a formal definition of our learning model.

Definition 12 (Testable Learning with Distribution Shift (TDS Learning))**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ and consider a distribution D over $\mathcal{X}$ and a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}\}$. For some $\psi : [0,1] \to [0,1]$ and $\epsilon, \delta \in (0,1)$, we say that an algorithm $\mathcal{A}$ testably learns $\mathcal{C}$ with distribution shift w.r.t. D up to error $\psi(\lambda) + \epsilon$ and probability of failure δ if the following is true. For any distributions $\mathcal{D}^{\text{train}}_{\mathcal{XY}}, \mathcal{D}^{\text{test}}_{\mathcal{XY}}$ over $\mathcal{X} \times \{\pm 1\}$ such that $\mathcal{D}^{\text{train}}_{\mathcal{X}} = D$, algorithm $\mathcal{A}$, upon receiving a large enough set of labelled samples S_{train} from the training distribution $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ and a large enough set of unlabelled samples X_{test} from the test distribution $\mathcal{D}^{\text{test}}_{\mathcal{X}}$, either rejects $(S_{\text{train}}, X_{\text{test}})$ or accepts and outputs a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ with the following guarantees.*

(*a*) *(Soundness.) With probability at least $1 - \delta$ over the samples $S_{\text{train}}, X_{\text{test}}$ we have:*
If A accepts, then the output h satisfies $\operatorname{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) \leq \psi(\lambda) + \epsilon$.

(*b*) *(Completeness.) Whenever $\mathcal{D}^{\text{test}}_{\mathcal{X}} = \mathcal{D}^{\text{train}}_{\mathcal{X}}$, A accepts with probability at least $1 - \delta$ over the samples $S_{\text{train}}, X_{\text{test}}$.*

In particular, we say that $\mathcal{A}$ testably learns $\mathcal{C}$ with distribution shift w.r.t. D in the realizable setting, if $\mathcal{A}$ is required to satisfy the above guarantees only when $\mathcal{D}^{\text{train}}_{\mathcal{XY}}, \mathcal{D}^{\text{test}}_{\mathcal{XY}}$ and $\mathcal{C}$ are realizable (where $\lambda = 0 = \psi(\lambda)$).

3.7 TDS Learning of Homogeneous Halfspaces

We now provide a proof of Theorem 13, which we restate here for convenience.

Theorem 19 (Agnostic TDS learning of Halfspaces)**.** *Let $\mathcal{C}$ be the class of origin-centered halfspaces over $\mathbb{R}^d$ and $C > 0$ a sufficiently large universal constant. Let $\mathcal{A}, \mathcal{T}$ be as defined in Propositions 32 and 33. Let $m_{\mathcal{A}}$ be the sample complexity of $\mathcal{A}(\epsilon/C, \delta/4)$ and $m_{\mathcal{T}} = \frac{Cd^4}{\epsilon^2 \delta}$. Then, Algorithm 1, given inputs $S_{\text{train}}, X_{\text{test}}$ of sizes $|S_{\text{train}}| \geq m_{\mathcal{A}}$ and $|X_{\text{test}}| \geq m_{\mathcal{T}}$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. any isotropic log-concave distribution D with error $O(\lambda) + \epsilon$ and run-time $\operatorname{poly}(d, \frac{1}{\epsilon}) \log\left(\frac{1}{\delta}\right)$, where ϵ is the accuracy parameter and δ is the failure probability.*

Algorithm 1: Agnostic TDS Learning of Halfspaces

Input: Sets S_{train} from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$, X_{test} from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$, parameters $\epsilon > 0$, $\delta \in (0,1)$
Set $\epsilon' = \epsilon/C$ where C is some sufficiently large universal constant.
Let $m_{\mathcal{A}}$ be the sample complexity of $\mathcal{A}(\epsilon', \delta/4)$.
Split S_{train} to S_1, S_2 with sizes $m_{\mathcal{A}}$, $\frac{C}{\epsilon^2}\log(1/\delta)$
Run $\mathcal{A}(\epsilon', \delta/4)$ on S_1 and obtain $\widehat{\mathbf{v}} \in \mathbb{S}^{d-1}$
Let $\widehat{\epsilon} = \mathbb{P}_{(\mathbf{x},y)\sim S_2}[\mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq y]$.
Run $\mathcal{T}(\widehat{\epsilon}, \delta/2)$ on X_{test}.
Reject and terminate if $\mathcal{T}$ rejects.
Otherwise, output $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$ with $\widehat{f} : \mathbf{x} \to \mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x})$.

In order to prove the above theorem, we make use of the following agnostic learning result from [DKTZ20b].

Proposition 33 (Theorem 3.1 in [DKTZ20b])**.** *Let $\mathcal{D}_{\mathcal{XY}}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ such that its marginal on $\mathbb{R}^d$ is isotropic log-concave. Then there is an algorithm $\mathcal{A}$ such that for any $\epsilon > 0$ and $\delta \in (0,1)$, $\mathcal{A}(\epsilon, \delta)$, upon drawing $m = \tilde{O}(\frac{d}{\epsilon^4}\log(1/\delta))$ independent examples from $\mathcal{D}_{\mathcal{XY}}$ and in time $\mathrm{poly}(d, 1/\epsilon) \cdot \log(1/\delta)$, outputs $\widehat{\mathbf{v}} \in \mathbb{S}^{d-1}$ such that, with probability at least $1-\delta$, the corresponding halfspace has error at most $O(\eta) + \epsilon$, where η is the error of the optimal halfspace on $\mathcal{D}_{\mathcal{XY}}$.*

We also use the following fact about isotropic log-concave distributions.

Fact 1. $\mathbb{P}_{\mathbf{x}\sim D}[\mathit{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \mathit{sign}(\mathbf{v}^* \cdot \mathbf{x})] = \Theta(\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*))$*, when D is isotropic log-concave.*

Proof. Suppose that S_{train} is a set of m_{train} independent samples from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$, where the marginal of $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ on $\mathbb{R}^d$ is the standard Gaussian distribution. Let also X_{test} be a set of m_{test} independent unlabelled samples from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$. In what follows, let $\epsilon' = \epsilon/C$ and let $C > 0$ be a sufficiently large universal constant. Let also $m_{\mathcal{A}}$ be the sample complexity of $\mathcal{A}(\epsilon', \delta/4)$ and $m_{\mathcal{T}} = \frac{Cd^4}{\epsilon^2\delta}$.

Soundness. Suppose that the algorithm accepts. Let $\mathbf{v}^* \in \mathbb{S}^{d-1}$ define the halfspace f^* that achieves $\mathrm{err}(f^*; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) + \mathrm{err}(f^*; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) = \lambda$. Note that since $|S_2| \geq \frac{C}{\epsilon^2}\log(1/\delta)$, we have that $\widehat{\epsilon} \leq \mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) + \epsilon'$. By Proposition 33, since $|S_1| \geq m_{\mathcal{A}}$ we have $\mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) \leq \eta + \epsilon'$, where $\eta \in (0,1)$ is the error of the optimum halfspace, say $f : \mathbf{x} \mapsto \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})$ on $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$. Note that $\eta \leq \lambda$. We have that $\mathrm{err}(\widehat{f}, f; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}) \leq \mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) + \mathrm{err}(f; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) \leq 2\eta + \epsilon'$. Therefore, due to Fact 1, and since $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}} = D$, we obtain $\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}) \leq 2C'\eta + C'\epsilon'$ for some sufficiently large $C' > 0$ (with $C \gg C'$).

Moreover, we have that $\mathrm{err}(f^*; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) \leq \lambda$ and, hence $\mathrm{err}(f^*, f; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}) \leq \lambda + C'\eta$. We now apply Proposition 32, to obtain $\mathrm{err}(\widehat{f}, f^*; X_{\mathrm{test}}) \leq \sqrt{C}\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*)$. Since $|X_{\mathrm{test}}| \geq \frac{\sqrt{C}}{\epsilon^2}\log(1/\delta)$, due to standard VC dimension arguments, we have $\mathrm{err}(\widehat{f}, f^*; \mathcal{D}^{\mathrm{test}}_{\mathcal{X}}) \leq \sqrt{C}\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*) + \epsilon'$. By Fact 1,

$\measuredangle(\widehat{\mathbf{v}}, \mathbf{v}^*) \le C'\mathrm{err}(\widehat{f}, f^*; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}})$. Therefore, with probability at least $1-\delta$, we have

$$\begin{aligned}\mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) &\le \mathrm{err}(\widehat{f}, f^*; \mathcal{D}^{\mathrm{test}}_{\mathcal{X}}) + \mathrm{err}(f^*; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \le C'\sqrt{C}\mathrm{err}(\widehat{f}, f^*; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}) + \epsilon' + \lambda \\ &\le C'\sqrt{C}\mathrm{err}(\widehat{f}, f; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}) + C'\sqrt{C}\mathrm{err}(f, f^*; \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}) + \epsilon' + \lambda \\ &\le C\lambda + C\epsilon' \le \epsilon\end{aligned}$$

Completeness. Readily follows from Proposition 32 and $|X_{\mathrm{test}}| \ge m_{\mathcal{T}}$. □

Remark 2. *We note that, in fact, the original version of Proposition 32 in Chapter 2 does not require the target marginal to be known, but works universally for any isotropic log-concave distribution (as well as distributions with heavier tails). This implies that the completeness criterion that Algorithm 1 satisfies is actually much stronger: for an appropriate choice of the absolute constant C, Algorithm 1 can be made to accept whenever $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ is isotropic log-concave (and not necessarily equal to the training marginal).*

Remark 3. *Moreover, we point out that we can apply results from Chapter 2 and substitute algorithm $\mathcal{A}$ with a universal tester-learner for halfspaces. This enables us to remove the assumption that $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}$ is some fixed isotropic log-concave distribution, and the final algorithm would accept with high probability whenever $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}$ is isotropic strongly log-concave* and *$\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ is isotropic log-concave. In that sense, TDS learning composes well with (universally) testable learning. For sake of presentation, however, we leave formal compositional arguments to future work.*

3.8 Realizable TDS Learning

3.8.1 Disagreement-Based TDS Learners

In this section, we prove Theorem 14. First, we prove the following a special version regarding realizable TDS learning of homogeneous halfspaces with respect to the Gaussian distribution.

Proposition 34 (TDS learning of Homogeneous Halfspaces)**.** *Let $\mathcal{C}$ be the class of origin-centered halfspaces over $\mathbb{R}^d$ and $C > 0$ a sufficiently large universal constant. Then, Algorithm 2, given inputs $S_{\mathrm{train}}, X_{\mathrm{test}}$ of sizes $|S_{\mathrm{train}}| \ge C(\frac{d}{\epsilon})^{3/2}\log\big(\frac{1}{\epsilon\delta}\big)$ and $|X_{\mathrm{test}}| \ge C\frac{d}{\epsilon^2}\log\big(\frac{1}{\epsilon\delta}\big)$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. the standard Gaussian distribution $\mathcal{N}(0, I_d)$ with run-time $\mathrm{poly}(d, 1/\epsilon)\log\big(\frac{1}{\delta}\big)$, where ϵ is the accuracy parameter and δ is the failure probability.*

We will use the following fact about the Gaussian distribution.

Fact 2. *For any $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{S}^{d-1}$ we have $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\mathit{sign}(\mathbf{v}_1\cdot\mathbf{x}) \ne \mathit{sign}(\mathbf{v}_2\cdot\mathbf{x})] = \measuredangle(\mathbf{v}_1, \mathbf{v}_2)/\pi$.*

Proof of Proposition 34. Suppose that S_{train} is a set of m_{train} independent samples from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$, where the marginal of $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ on $\mathbb{R}^d$ is the standard Gaussian distribution. Let also X_{test} be a set of m_{test} independent unlabelled samples from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$. In what follows, let $\epsilon' = \epsilon^{3/2}/(8d^{1/2})$.

Algorithm 2: TDS Learning of Homogeneous Halfspaces

Input: Sets S_{train} from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$, X_{test} from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$, parameter $\epsilon > 0$

Set $\epsilon' = \epsilon^{3/2}/(10d^{1/2})$.

Run the Empirical Risk Minimization algorithm on S_{train} up to error ϵ', i.e., compute a vector $\widehat{\mathbf{v}} \in \mathbb{S}^{d-1}$ with $\widehat{\mathbf{v}} = \arg\min_{\mathbf{v}' \in \mathbb{S}^{d-1}} \mathbb{P}_{(\mathbf{x},y)\in S_{\mathrm{train}}}[y \neq \mathrm{sign}(\mathbf{v}' \cdot \mathbf{x})]$

Let $\mathcal{V} = \{\mathbf{v}' \in \mathbb{S}^{d-1} : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \epsilon'\}$.

For each $\mathbf{x} \in X_{\mathrm{test}}$, compute the following quantities.

$$\mathbf{v}^+_{\mathbf{x}} = \arg\max_{\mathbf{v}' \in \mathcal{V}} \mathbf{v}' \cdot \mathbf{x} \text{ and } \mathbf{v}^-_{\mathbf{x}} = \arg\min_{\mathbf{v}' \in \mathcal{V}} \mathbf{v}' \cdot \mathbf{x}$$

Reject and terminate if $\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})] > 3\epsilon/4$.

Otherwise, output $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$ with $\widehat{f} : \mathbf{x} \mapsto \mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x})$.

Soundness. When the algorithm accepts, we have that $\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})] \leq \frac{3\epsilon}{2}$. By standard VC dimension arguments and Fact 2, after running the Empirical Risk Minimization algorithm on training data, as long as $m_{\mathrm{train}} \geq C\frac{d}{\epsilon'}\log(1/(\delta\epsilon'))$, we have $\|\widehat{\mathbf{v}} - \mathbf{v}\|_2 \leq \epsilon'$. Therefore, both $\mathbf{v}$ and $\widehat{\mathbf{v}}$ are within $\mathcal{V} = \{\mathbf{v}' \in \mathbb{S}^{d-1} : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \epsilon'\}$. By the definition of $\mathbf{v}^+_{\mathbf{x}}$ and $\mathbf{v}^-_{\mathbf{x}}$, we have the following.

$$\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})] \leq \mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})] \leq 3\epsilon/4 \tag{3.3}$$

Moreover, we have $\mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) = \mathbb{E}[\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]]$, where the expectation is over $X_{\mathrm{test}} \sim (\mathcal{D}^{\mathrm{test}}_{\mathcal{X}})^{\otimes m_{\mathrm{test}}}$. By standard VC dimension arguments, we have that, with probability at least $1 - \delta/2$, $\mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) = \mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})] + \epsilon/4$ whenever $m_{\mathrm{test}} \geq C\frac{d}{\epsilon^2}\log(1/(\delta\epsilon))$. Therefore, with probability at least $1 - \delta$ (union bound over two bad events), upon acceptance, we have $\mathrm{err}(\widehat{f}; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \leq \epsilon$.

Completeness. For completeness, we assume that X_{test} is drawn from $\mathcal{N}(0, I_d)$. Observe that $\mathcal{V}$ does not depend on X_{test} (since it is formed only using training data). Therefore, we may apply a standard Hoeffding bound to ensure that with probability at least $1 - \delta$, whenever $m_{\mathrm{test}} \geq C\frac{1}{\epsilon^2}\log(1/(\delta))$, we have

$$\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})] \leq \mathbb{P}_{\mathbf{x}\sim \mathcal{N}(0,I_d)}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})] + \epsilon/4$$

It remains to bound $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})]$ by $\epsilon/2$. We observe that, since $\mathbf{v}^+, \mathbf{v}^- \in \mathcal{V}$, we have $\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x} \geq \mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x} - \|\mathbf{v}^+_{\mathbf{x}} - \mathbf{v}^-_{\mathbf{x}}\|_2\|\mathbf{x}\|_2 \geq \mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x} - \epsilon'\|\mathbf{x}\|_2 \geq \widehat{\mathbf{v}} \cdot \mathbf{x} - \epsilon'\|\mathbf{x}\|_2$ by the definition of $\mathbf{v}^+_{\mathbf{x}}$ and $\mathbf{v}^-_{\mathbf{x}}$. We similarly have $\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x} \leq \widehat{\mathbf{v}} \cdot \mathbf{x} + \epsilon'\|\mathbf{x}\|_2$.

Therefore the probability that $\mathrm{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x}) \neq \mathrm{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x})$ is upper bounded by the probability

that $|\widehat{\mathbf{v}} \cdot \mathbf{x}| \leq \epsilon' \|\mathbf{x}\|_2$ (since, otherwise, both $\mathbf{v}_{\mathbf{x}}^+ \cdot \mathbf{x}$ and $\mathbf{v}_{\mathbf{x}}^- \cdot \mathbf{x}$ have the same sign). In particular

$$\begin{aligned}
\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\operatorname{sign}(\mathbf{v}_{\mathbf{x}}^+ \cdot \mathbf{x}) \neq \operatorname{sign}(\mathbf{v}_{\mathbf{x}}^- \cdot \mathbf{x})] &\leq \mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[|\widehat{\mathbf{v}} \cdot \mathbf{x}| \leq \epsilon' \|\mathbf{x}\|_2] \\
&\leq \mathbb{P}_{\mathbf{x}\sim\mathcal{N}_d}[\|\mathbf{x}\|_2 > \sqrt{4d/\epsilon}] + \mathbb{P}_{\mathbf{x}\sim\mathcal{N}_d}[|\widehat{\mathbf{v}} \cdot \mathbf{x}| \leq \epsilon' \sqrt{4d/\epsilon}] \\
&\leq \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{N}_d}[\|\mathbf{x}\|_2^2]\epsilon}{4d} + \mathbb{P}_{\mathbf{x}\sim\mathcal{N}_d}[|\widehat{\mathbf{v}} \cdot \mathbf{x}| \leq \epsilon' \sqrt{4d/\epsilon}]
\end{aligned}$$

We obtain the final inequality by applying Markov's inequality. Since $\mathbb{E}_{\mathbf{x}\sim\mathcal{N}_d}[\|\mathbf{x}\|_2^2] = d$ and the one-dimensional Gaussian density is upper bounded by $(2\pi)^{-1}$, we have the following bound.

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\operatorname{sign}(\mathbf{v}_{\mathbf{x}}^+ \cdot \mathbf{x}) \neq \operatorname{sign}(\mathbf{v}_{\mathbf{x}}^- \cdot \mathbf{x})] \leq \frac{\epsilon}{4} + \frac{2}{\sqrt{2\pi}}\epsilon' \sqrt{4d/\epsilon} \leq \epsilon/2\,,$$

since $\epsilon' \leq \epsilon^{3/2}/(8d^{1/2})$. This completes the proof. □

We now prove Theorem 14, which we restate here for convenience.

Theorem 20 (Disagreement-Based TDS learning)**.** *Let $\mathcal{C}$ be the class of concepts that map $\mathcal{X} \subseteq \mathbb{R}^d$ to $\{\pm 1\}$ with VC dimension $\mathrm{VC}(\mathcal{C})$, let D a distribution over $\mathcal{X}$ and $C > 0$ a sufficiently large universal constant. Suppose that we have access to an ERM oracle for PAC learning $\mathcal{C}$ under D and membership access to $\mathbf{D}_{\epsilon'}(f; D)$ for any given $f \in \mathcal{C}$ and $\epsilon' > 0$. Then, Algorithm 3, given inputs of sizes $|S_{\mathrm{train}}| \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon'}\log\left(\frac{1}{\epsilon'\delta}\right)$ and $|X_{\mathrm{test}}| \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon^2}\log\left(\frac{1}{\epsilon\delta}\right)$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. D that calls the ϵ'-ERM oracle once and the ϵ'-membership oracle $|S_{\mathrm{train}}|$ times, where ϵ is the accuracy parameter, δ is the failure probability and ϵ' such that $\epsilon' \cdot \theta(\epsilon', d) \leq \epsilon/2$.*

Algorithm 3: Disagreement-Based TDS Learning

Input: Sets S_{train} from $\mathcal{D}_{\mathcal{XY}}^{\mathrm{train}}$, X_{test} from $\mathcal{D}_{\mathcal{X}}^{\mathrm{test}}$, parameter $\epsilon > 0$
Set $\epsilon' > 0$ such that $\epsilon' \cdot \theta(\epsilon', d) \leq \epsilon/2$.
Run the Empirical Risk Minimization algorithm on S_{train} up to error ϵ', i.e., compute $\widehat{f} \in \mathcal{C}$ with $\widehat{f} = \arg\min_{f' \in \mathcal{C}} \mathbb{P}_{(\mathbf{x},y)\in S_{\mathrm{train}}}[y \neq f'(\mathbf{x})]$
Let $\mathbf{D}_{\epsilon'}(\widehat{f}; D)$ be as in Definition 10.
Reject and terminate if $\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(\widehat{f}; D)] > \epsilon/2$.
Otherwise, output $\widehat{f}$.

Proof of Theorem 14. Suppose that S_{train} is a set of m_{train} independent samples from $\mathcal{D}_{\mathcal{XY}}^{\mathrm{train}}$, where the marginal of $\mathcal{D}_{\mathcal{XY}}^{\mathrm{train}}$ on $\mathcal{X}$ is the distribution D. Let also X_{test} be a set of m_{test} independent unlabelled samples from $\mathcal{D}_{\mathcal{X}}^{\mathrm{test}}$. In what follows, let $\epsilon' > 0$ such that $\epsilon'\theta(\epsilon', d) \leq \epsilon/2$. The proof follows a similar recipe as the one of Proposition 34. For the following, let $f^* \in \mathcal{C}$ be the label generating function.

Soundness. Suppose that the algorithm accepts. Then, $\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(\widehat{f}; D)] \leq \epsilon/2$. Since $\widehat{f}$ is an minimizes the empirical error on training data, by standard VC arguments, we have that $\mathrm{err}(\widehat{f}, f^*; D) \leq \epsilon/2$, whenever $m_{\mathrm{train}} \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon'}\log\big(\frac{1}{\epsilon'\delta}\big)$, since $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}} = D$ by assumption. Therefore, by the definition of $\mathbf{D}_{\epsilon'}(\widehat{f}; D)$, for any $\mathbf{x} \notin \mathbf{D}_{\epsilon'}(\widehat{f}; D)$, we have $\widehat{f}(\mathbf{x}) = f^*(\mathbf{x})$. Therefore, we have

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim X_{\mathrm{test}}}[\widehat{f}(\mathbf{x}) \neq f^*(\mathbf{x})] \leq \mathop{\mathbb{P}}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(\widehat{f}; D)] \leq \epsilon/2$$

Whenever $m_{\mathrm{test}} \geq C\frac{\mathrm{VC}(\mathcal{C})}{\epsilon^2}\log\big(\frac{1}{\epsilon\delta}\big)$, we have $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}}[y \neq f^*(\mathbf{x})] \leq \mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\widehat{f}(\mathbf{x}) \neq f^*(\mathbf{x})] + \epsilon/2 \leq \epsilon$.

Completeness. Suppose that $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}} = D$. Then, by a standard Hoeffding bound, we have that whenever $m_{\mathrm{test}} \geq C\frac{1}{\epsilon}\log(1/\delta)$, we have $\mathbb{P}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathbf{x} \in \mathbf{D}_{\epsilon'}(\widehat{f}; D)] \leq \mathbb{P}_{\mathbf{x}\sim D}[\mathbf{D}_{\epsilon'}(\widehat{f}; D)] + \epsilon/2$ with probability at least $1-\delta$ and $\mathbb{P}_{\mathbf{x}\sim D}[\mathbf{D}_{\epsilon'}(\widehat{f}; D)] \leq \epsilon'\theta(\epsilon', d) \leq \epsilon/2$, by the choice of ϵ'. □

3.8.2 TDS Learner for General Halfspaces

We now prove Theorem 15 which we restate here for convenience.

Theorem 21 (TDS learning of General Halfspaces)**.** *Let $\mathcal{C}$ be the class of general halfspaces over $\mathbb{R}^d$ and $C > 0$ a sufficiently large universal constant. Then, Algorithm 4, given inputs of size $|S_{\mathrm{train}}| = |X_{\mathrm{test}}| = Cd^{C\log 1/\epsilon}$ is a TDS learning algorithm for $\mathcal{C}$ w.r.t. the standard Gaussian distribution $\mathcal{N}(0, I_d)$ with run-time $d^{O(\log 1/\epsilon)}$, where ϵ is the accuracy parameter, and the failure probability δ is at most* 0.01.

Suppose the ground-truth halfspace $f^*(\mathbf{x}) = \mathrm{sign}(\mathbf{x}\cdot\mathbf{v} - \tau)$ is determined by a unit vector $\mathbf{v} \in \mathbb{R}^d$ and a value $\tau \in \mathbb{R}$. We will need the following showing that if a halfspace not too biased under the standard Gaussian distribution, then it is possible to recover the parameters of the halfspace up to a very high accuracy. See Subsection 3.8.2 for the proof.

Proposition 35 (Parameter recovery for halfspaces)**.** *For a sufficiently large absolute constant $C > 0$, the following is true. For every $\beta, \gamma \in (0,1)$ and integer d, let S_{train} be a set of $C(\frac{d}{\beta\gamma})^C$ i.i.d samples from a distribution $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ such that $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}} = \mathcal{N}(0, I_d)$ and the labels are given by an unknown halfspace $f : \mathbf{x} \mapsto$ sign$(\mathbf{v}\cdot\mathbf{x} - \tau)$. Additionally, assume that the halfspace f satisfies $\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[f^*(\mathbf{x}) = -1] \geq \gamma$ and $\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[f^*(\mathbf{x}) = 1] \geq \gamma$. Let $\mathcal{T} = \{\widehat{\mathbf{v}}\cdot\mathbf{x} : (\mathbf{x}, y) \in S_{\mathrm{train}}\}$ and set*

$$\widehat{\mathbf{v}} = \frac{\sum_{(\mathbf{x},y)\in S_{\mathrm{train}}}\mathbf{x}y}{\left\|\sum_{(\mathbf{x},y)\in S_{\mathrm{train}}}\mathbf{x}y\right\|_2} \text{ and } \widehat{\tau} = \mathop{\arg\min}_{\tau'\in\mathcal{T}} \mathop{\mathbb{P}}_{(\mathbf{x},y)\in S_{\mathrm{train}}}[f^*(\mathbf{x}) \neq \mathit{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x} - \tau')].$$

Then, with probability at least $1 - 1/1000$ we have $\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \leq \beta$ and $|\tau - \widehat{\tau}| \leq \beta$.

We also highlight two technical lemmas that we use for the analysis of Algorithm 4. Our first technical lemma insures that if f is a halfspace that very likely assigns the same label to samples

Algorithm 4: TDS Learning of General Halfspaces

Input: Sets S_{train} from $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$, X_{test} from $\mathcal{D}^{\text{test}}_{\mathcal{X}}$, parameter $\epsilon > 0$

Set $T = 2^{C_1^2 \log \frac{1}{\epsilon}+1}$, $k = C_1 \log \frac{1}{\epsilon}$, $\Delta = \frac{\epsilon}{d^{C_2 k}}$ and $\beta = \frac{\epsilon^2}{C_3 d^{C_3}}$.

if $\mathbb{P}_{(\mathbf{x},y)\sim S_{\text{train}}}[y \neq b] \leq \frac{1}{T}$ for some $b \in \{\pm 1\}$ (large bias case) **then**

  For each $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq k$, compute the quantity $\widehat{\mathrm{M}}_\alpha = \mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}^\alpha]$.

  Reject and terminate if $|\widehat{\mathrm{M}}_\alpha - \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\mathbf{x}^\alpha]| > \Delta$ for some α with $\|\alpha\|_1 \leq k$.

  Otherwise, output $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$ and terminate, where $\widehat{f} : \mathbf{x} \mapsto b$ ($\widehat{f}$ constant).

else

  Set $\widehat{\mathbf{v}} = \frac{\mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}[y\mathbf{x}]}{\|\mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}[y\mathbf{x}]\|_2}$.

  Let $\mathcal{T} = \{\widehat{\mathbf{v}} \cdot \mathbf{x} : (\mathbf{x}, y) \in S_{\text{train}}\}$.

  Set $\widehat{\tau} = \arg\min_{\tau\in\mathcal{T}} \mathbb{P}_{(\mathbf{x},y)\in S_{\text{train}}}[f^*(\mathbf{x}) \neq \text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x} - \tau')]$,

  Let $\mathcal{V} = \{(\mathbf{v}', \tau') : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \beta, |\tau' - \widehat{\tau}| \leq \beta\}$.

  For each $\mathbf{x} \in X_{\text{test}}$, compute the following quantities.

$$(\mathbf{v}^+_{\mathbf{x}}, \tau^+_{\mathbf{x}}) = \arg\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}' \cdot \mathbf{x} - \tau' \text{ and } (\mathbf{v}^-_{\mathbf{x}}, \tau^-_{\mathbf{x}}) = \arg\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}' \cdot \mathbf{x} - \tau'$$

  Reject and terminate if $\mathbb{P}_{\mathbf{x}\sim X_{\text{test}}}[\text{sign}(\mathbf{v}^+_{\mathbf{x}} \cdot \mathbf{x} - \tau^+_{\mathbf{x}}) \neq \text{sign}(\mathbf{v}^-_{\mathbf{x}} \cdot \mathbf{x} - \tau^-_{\mathbf{x}})] > 10\epsilon$.

  Otherwise, output $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$ with $\widehat{f} : \mathbf{x} \mapsto \text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x} - \widehat{\tau})$.

end if

from the Gaussian distribution, then f also very likely assigns the same label to samples form a distribution whose low-degree moments match those of a Gaussian. This lemma will be useful for proving the soundness of Algorithm 4, and is proven in Section 3.8.2. (Recall that for $\mathbf{x} \in \mathbb{R}^d$ we denote $\prod_{i=1}^n x_i^{\alpha_i}$ as $\mathbf{x}^\alpha$.)

Lemma 14. *When C_1 and C_2 both exceed some specific absolute constant, the following holds. Let k and T be defined as in Algorithm 4. Suppose, the set X_{test} is such that for every collection of non-negative integers $(\alpha_1, \cdots, \alpha_d)$ satisfying $\sum_i \alpha_i \leq k$ we have*

$$\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim X_{\text{test}}} [\mathbf{x}^\alpha] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)} [\mathbf{x}^\alpha] \right| \leq \frac{\epsilon}{d^{C_2 k}}. \tag{3.4}$$

Also, suppose the function $f^*(\mathbf{x}) = \text{sign}(\mathbf{x} \cdot \mathbf{v} - \tau)$ and the value $L \in \{\pm 1\}$ are such that

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,1)}[f^*(\mathbf{x}) \neq L] \leq \frac{2}{T}. \tag{3.5}$$

Then, it is the case that

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim X_{\text{test}}}[f^*(\mathbf{x}) \neq L] \leq O(\epsilon). \tag{3.6}$$

Our second technical lemma bounds, for $\mathbf{x}$ chosen from the standard Gaussian, the probability that one is unsure about $f^*(\mathbf{x}) = \text{sign}(\mathbf{v}\cdot\mathbf{x}-\tau)$ when one only has approximate estimates for $\widehat{\mathbf{v}}$ and

$\widehat{\tau}$ for $\mathbf{v}$ and τ respectively. This lemma will be useful for proving the completeness of Algorithm 4, and is proven in Section 3.8.2.

Lemma 15. *There is some absolute constant K_1, such that for every positive integer d and $\beta \in (0,1)$, the following holds. Let $\widehat{\mathbf{v}}$ be any unit vector in $\mathbb{R}^d$ and $\widehat{\tau}$ be in $\mathbb{R}$. Then, we have for $\mathcal{V} = \{(\mathbf{v}', \tau') : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \beta, |\tau' - \widehat{\tau}| \leq \beta\}$*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[sign\left(\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right) \neq sign\left(\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right)\right] \leq K_1 d^{K_1}\sqrt{\beta} \tag{3.7}$$

Proof of Soundness.

In this subsection we show that if Algorithm 4 accepts then the output $\widehat{f}$ of our algorithm will generalize on the distribution $\mathcal{D}_{\mathcal{X}}^{\text{test}}$.

Proposition 36 (Soundness)**.** *For any sufficiently large absolute constant C, the following is true. For any distribution $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ and any halfspace $f = sign(\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})$, the following is true. It can happen with probability only at most $\frac{1}{100}$ that Algorithm 4 gives an output (ACCEPT, $\widehat{f}$) for some predictor $\widehat{f}$, but it is not the case that*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}^{\text{test}}}[f^*(\mathbf{x}) \neq \widehat{f}(\mathbf{x})] \leq O(\epsilon).$$

To prove this proposition, we first need to prove Lemma 14.

Proof of Lemma 14. First of all, we claim that Equation 3.5 implies that

$$|\tau| \geq \sqrt{\frac{1}{2}\log\frac{T}{2}} \tag{3.8}$$

Indeed, we have

$$\frac{2}{T} \geq \frac{1}{\sqrt{2\pi}}\int_{|\tau|}^{\infty} e^{-z^2/2}dz \geq |\tau|e^{-2|\tau|^2} \geq e^{-2|\tau|^2},$$

where the last inequality holds because for sufficiently large C_1 the value of T and therefore $|\tau|$ is sufficiently large and exceeds 1.

Recall that $\mathbf{v}$ is assumed to be a unit vector in $\mathbb{R}^d$. Assume, without loss of generality, that $L = -1$, and therefore $\tau > 0$. We have

$$\mathbb{P}_{\mathbf{x}\sim X_{\text{test}}}[\text{sign}(\mathbf{x}\cdot\mathbf{v} - \tau) \neq -1] = \mathbb{P}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}\cdot\mathbf{v} \geq \tau] \leq \frac{\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[(\mathbf{x}\cdot\mathbf{v})^k]}{\tau^k}. \tag{3.9}$$

To use this inequality, we need to upper-bound $\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[(\mathbf{x}\cdot\mathbf{v})^k]$. Since $\mathbf{v}$ is a unit vector, every (of at most d^k) terms of the polynomial mapping $\mathbf{x}\in\mathbb{R}^d$ to $(\mathbf{x}\cdot\mathbf{v})^k$ has coefficient at most 1. This,

together with Equation 3.4 and the triangle inequality, allows us to conclude that

$$\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim X_{\text{test}}}[(\mathbf{x}\cdot\mathbf{v})^k] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[(\mathbf{x}\cdot\mathbf{v})^k] \right| \leq d^k \frac{\epsilon}{d^{C_2 k}}.$$

Now, since $\mathbf{v}$ is a unit vector, we have $\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[(\mathbf{x}\cdot\mathbf{v})^k] = k!! \leq k^k$. Combining this with the equation above, and Equation 3.9 and then substituting Equation 3.8 and the values of k and T we get:

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim X_{\text{test}}}[\text{sign}(\mathbf{x}\cdot\mathbf{v}-\tau)\neq -1] \leq \frac{1}{|\tau|^k}\left(k^{k/2} + d^k\frac{\epsilon}{d^{C_2 k}}\right) \leq \frac{1}{\left(\frac{1}{2}C_1^2\log\frac{1}{\epsilon}\right)^{C_1\log\frac{1}{\epsilon}}}\left(\left(C_1\log\frac{1}{\epsilon}\right)^{C_1\log\frac{1}{\epsilon}} + d^k\frac{\epsilon}{d^{C_2 k}}\right)$$

We see that when C_1 and C_2 both exceed some absolute constant, the above expression is at most ϵ, which completes the proof. □

Having proven Lemma 14, we are now ready to prove Proposition 36.

Proof of Proposition 36. First, suppose the algorithm outputs (ACCEPT, L) for some $L \in \{\pm 1\}$ via Step 5. For the algorithm to reach this step, it has to be that

$$\mathop{\mathbb{P}}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L] \leq \frac{1}{T},$$

Via Hoeffding's inequality, if C is sufficiently large then with probability at least $1 - \frac{1}{1000}$ it holds that

$$|\mathop{\mathbb{P}}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L] - \mathop{\mathbb{P}}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L]| \leq \frac{1}{2T}, \tag{3.10}$$

and combining the two equations above

$$\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[f^*(\mathbf{x}) \neq L] \leq \frac{2}{T}.$$

Furthermore, for the algorithm not output REJECT in Step 4, it has to be the case that for every collection of non-negative integers $(\alpha_1, \cdots, \alpha_d)$ satisfying $\sum_i \alpha_i \leq k$ we have

$$\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}^\alpha] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\mathbf{x}^\alpha] \right| > \frac{\epsilon}{d^{C_2 k}}.$$

Overall, this allows us to apply Lemma 14 to conclude that

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim X_{\text{test}}}[f^*(\mathbf{x}) \neq L] \leq O(\epsilon),$$

and, for a sufficiently large absolute constant C, with probability at least $1 - \frac{1}{1000}$, this is only

possible if

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}^{\text{test}}}[f^*(\mathbf{x}) \neq L] \leq O(\epsilon),$$

which finishes the proof for the case when the algorithm accepts in Step 5.

Now, suppose the algorithm accepts in Step 13. For the algorithm to reach this step, it has to be that

$$\mathbb{P}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L] > \frac{1}{T},$$

And together with Equation 3.10, this implies that

$$\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[f^*(\mathbf{x}) \neq L] > \frac{1}{2T}.$$

For such f^* we can apply Proposition 35 and conclude that with probability at least $1 - 1/1000$ the values of $\widehat{\mathbf{v}}$ and τ obtained in Algorithm 4 satisfy

$$\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \leq \left(\frac{\epsilon}{C_3 d^{C_3}}\right)^2 = \beta, \tag{3.11}$$

$$|\tau - \widehat{\tau}| \leq \left(\frac{\epsilon}{C_3 d^{C_3}}\right)^2 = \beta, \tag{3.12}$$

where the last equality is by the definition of β. Now, since the algorithm did not reject in Step 12, it must be the case that the fraction of elements in X_{test} that satisfy $\text{sign}(\mathbf{v}_{\mathbf{x}}^+ \cdot \mathbf{x} - \tau_{\mathbf{x}}^+) \neq \text{sign}(\mathbf{v}_{\mathbf{x}}^- \cdot \mathbf{x} - \tau_{\mathbf{x}}^-)$ is at most 10ϵ. If C is a sufficiently large absolute constant, the standard Hoeffding inequality tells us that for this to happen with probability larger than $1/1000$ it has to be the case that

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}^{\text{test}}}\left[\text{sign}\left(\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right) \neq \text{sign}\left(\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right)\right] \leq 11\epsilon.$$

Whenever the event above occurs, since $\mathcal{V} = \{(\mathbf{v}', \tau') : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \beta, |\tau' - \widehat{\tau}| \leq \beta\}$ we can use Equations 3.11 and 3.12 to conclude $\text{sign}(\mathbf{v}\cdot\mathbf{x} - \tau) = \text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})$. Therefore,

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}^{\text{test}}}[\text{sign}(\mathbf{v}\cdot\mathbf{x} - \tau) \neq \text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})] \leq 11\epsilon$$

This completes the proof of soundness of Algorithm 4. □

Proof of Completeness.

The second proposition shows that if the testing distribution is the standard Gaussian, then the algorithm will likely accept. Together, propositions 36 and 37 yield Theorem 15.

Proposition 37 (Completeness)**.** *For sufficiently large value of the absolute constants C and C_3 and for any halfspace $f = \text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})$, suppose the testing distribution $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ is the standard*

Gaussian distribution. Then, with probability at least $1 - \frac{1}{100}$ *Algorithm 4 will accept, i.e. output (ACCEPT,* $\widehat{f}$*) for some* $\widehat{f}$*.*

To prove this proposition, we first need to prove Lemma 15.

Proof of Lemma 15. We have $\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\|\mathbf{x}\|_2^2] = d$. Therefore, by Markov's inequality, we have

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\|\mathbf{x}\|_2 > \frac{\sqrt{d}}{\sqrt{\beta}}\right] = \mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\|\mathbf{x}\|_2^2 > \frac{d}{\beta}\right] \leq \beta \tag{3.13}$$

Additionally, from the bound of $\frac{1}{\sqrt{2\pi}}$ on the density of standard Gaussian in one dimension, we get:

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[|\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau}| \leq 100\sqrt{\beta d} + \beta\right] \leq \frac{200\sqrt{\beta d} + 2\beta}{\sqrt{2\pi}} \tag{3.14}$$

If it holds that $\|\mathbf{x}\|_2 \leq \frac{\sqrt{d}}{\sqrt{\beta}}$, we have for every $\mathbf{v}'$ satisfying $\|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \beta$ and any τ' satisfying $|\tau' - \widehat{\tau}| \leq \beta$ that

$$|\mathbf{v}'\cdot\mathbf{x} - \tau' - (\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})| \leq \sqrt{d\beta} + \beta$$

Therefore, if it is also the case that $|\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau}| > 100\sqrt{\beta d} + \beta$, then we have

$$\operatorname{sign}(\mathbf{v}'\cdot\mathbf{x} - \tau') = \operatorname{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau})$$

This allows us to conclude that

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\operatorname{sign}\left(\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right) \neq \operatorname{sign}\left(\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right)\right] \leq$$
$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\|\mathbf{x}\|_2 > \frac{\sqrt{d}}{\sqrt{\beta}}\right] + \mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[|\widehat{\mathbf{v}}\cdot\mathbf{x} - \widehat{\tau}| \leq 100\sqrt{\beta d} + \beta\right] \leq \beta + \frac{200\sqrt{\beta d} + 2\beta}{\sqrt{2\pi}},$$

where in the end we substituted Equation 3.13 and Equation 3.14. Recalling that for $\beta \in (0,1)$ we have $\beta < \sqrt{\beta}$ and picking K_1 to be a sufficiently large absolute constant, our proposition follows from the inequality above. □

Having proven Lemma 15, we are now ready to prove Proposition 37.

Proof of Proposition 37. There are two ways for the algorithm to output REJECT: through Step 4 and through Step 12. We will argue neither takes place. From standard Gaussian concentration, if C is a sufficiently large absolute const ant, with probability at least $1 - \frac{1}{1000}$ the algorithm will not output REJECT in Step 4.

We now proceed to ruling out the possibility that the algorithm outputs REJECT in Step 12.

For the algorithm to reach step Step 12, it is necessary that

$$\mathbb{P}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L] > \frac{1}{T},$$

Via Hoeffding's inequality, if C is sufficiently large then with probability at least $1 - \frac{1}{1000}$ it holds that $|\mathbb{P}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L] - \mathbb{P}_{\mathbf{x}\in S}[f^*(\mathbf{x}) \neq L]| \leq \frac{1}{2T}$, which together with the equation above implies that

$$\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[f^*(\mathbf{x}) \neq L] > \frac{1}{2T}.$$

For such f^* we can apply Proposition 35 and conclude that with probability at least $1 - 1/1000$ the values of $\widehat{\mathbf{v}}$ and τ obtained in Algorithm 4 satisfy

$$\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \leq \left(\frac{\epsilon}{C_3 d^{C_3}}\right)^2 = \beta, \tag{3.15}$$

$$|\tau - \widehat{\tau}| \leq \left(\frac{\epsilon}{C_3 d^{C_3}}\right)^2 = \beta, \tag{3.16}$$

Recall that $\mathcal{V} = \{(\mathbf{v}', \tau') : \|\mathbf{v}' - \widehat{\mathbf{v}}\|_2 \leq \beta, |\tau' - \widehat{\tau}| \leq \beta\}$. The equation above together with Lemma 15 implies that

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\text{sign}\left(\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right) \neq \text{sign}\left(\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right)\right] \leq K_1 d^{K_1}\frac{\epsilon}{C_3 d^{C_3}} \leq \epsilon,$$

where the last inequality holds for sufficiently large value of C_3. Combining the inequality above with the standard Hoeffding bound and recalling that $\mathcal{D}_{\mathcal{X}}^{\text{test}} = \mathcal{N}(0, I_d)$, we see that with probability at least $1 - \frac{1}{1000}$,

$$\mathbb{P}_{\mathbf{x}\sim X_{\text{test}}}\left[\text{sign}\left(\max_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right) \neq \text{sign}\left(\min_{(\mathbf{v}',\tau')\in\mathcal{V}} \mathbf{v}'\cdot\mathbf{x} - \tau'\right)\right] \leq 2\epsilon,$$

In conclusion, we see that the inequality above implies that the algorithm does not output REJECT in Step 12. This completes our proof. □

Parameter recovery.

Here we prove Proposition 35, which was used in the proofs of Proposition 36 and Proposition 37, thereby finishing the proof of Theorem 15. Let us first recall the setting of Proposition 35. For a unit vector $\mathbf{v}$ in $\mathbb{R}^d$ and $\tau \in \mathbb{R}$ satisfying

$$\min\left(\mathbb{P}_{x\in\mathcal{N}(0,I_d)}[\mathbf{v}\cdot\mathbf{x} - \tau > 0], \mathbb{P}_{x\in\mathcal{N}(0,I_d)}[\mathbf{v}\cdot\mathbf{x} - \tau < 0]\right) \geq \eta,$$

S_{train} is a set of $C\left(\frac{d}{\eta\beta}\right)^C$ i.i.d samples from a distribution $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$ with $\mathcal{X}$-marginal distributed as standard Gaussian and $\mathcal{Y}$-marginal given by the halfspace $f = \text{sign}(\mathbf{v} \cdot \mathbf{x} - \tau)$. The absolute constant C is assumed to be sufficiently large. We let $\mathcal{T} = \{\widehat{\mathbf{v}} \cdot \mathbf{x} : (\mathbf{x}, y) \in S_{\text{train}}\}$ and set

$$\widehat{\mathbf{v}} = \frac{\sum_{(\mathbf{x},y)\in S_{\text{train}}} \mathbf{x}y}{\left\|\sum_{(\mathbf{x},y)\in S_{\text{train}}} \mathbf{x}y\right\|_2}$$

$$\widehat{\tau} = \arg\min_{\tau'\in\mathcal{T}} \mathop{\mathbb{P}}_{(\mathbf{x},y)\in S_{\text{train}}}[f^*(\mathbf{x}) \neq \text{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x} - \tau')].$$

We would like to prove that with probability at least $29/30$ we have

$$\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \leq \beta,$$

$$|\tau - \widehat{\tau}| \leq \beta.$$

The following proposition tells us that the first inequality above is likely to hold:

Proposition 38 (Recovery of normal vector for halfspaces)**.** *For a sufficiently large absolute constant C, and every $\eta, \beta \in (0, 1)$ and integer d, the following holds. Let S_{train} is a set of at least $C\left(\frac{d}{\eta\beta}\right)^C$ i.i.d samples from a distribution $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$ with $\mathcal{X}$-marginal distributed as standard Gaussian and $\mathcal{Y}$-marginal given by the halfspace $f = \text{sign}(\mathbf{v} \cdot \mathbf{x} - \tau)$. For every unit vector $\mathbf{v}$ in $\mathbb{R}^d$ and $\tau \in \mathbb{R}$ satisfying*

$$\min\left(\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau > 0], \mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau < 0]\right) \geq \eta,$$

The vector $\widehat{\mathbf{v}} = \frac{\sum_{(\mathbf{x},y)\in S_{\text{train}}} \mathbf{x}y}{\left\|\sum_{(\mathbf{x},y)\in S_{\text{train}}} \mathbf{x}y\right\|_2}$ with probability at least $1 - \frac{1}{2000}$ satisfies:

$$\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \leq \beta,$$

Once this stage is accomplished, the next proposition tells us that we can recover the offset τ.

Proposition 39 (Offset recovery for halfspaces)**.** *For a sufficiently large absolute constant C, and every $\eta, \gamma \in (0, 1)$ and integer d, the following holds. Let S_{train} is a set of at least $C\left(\frac{d}{\eta\gamma}\right)^C$ i.i.d samples from a distribution $\mathcal{D}_{\mathcal{XY}}^{\text{train}}$ with $\mathcal{X}$-marginal distributed as standard Gaussian and $\mathcal{Y}$-marginal given by the halfspace $f = \text{sign}(v \cdot \mathbf{x} - \tau)$. For every unit vector $\mathbf{v}$ in $\mathbb{R}^d$ and $\tau \in \mathbb{R}$ satisfying*

$$\min\left(\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau > 0], \mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau < 0]\right) \geq \eta,$$

Then, with probability at least $1 - \frac{1}{2000}$, for every unit vector $\widehat{\mathbf{v}}$ that forms an angle of at most γ

with $\mathbf{v}$ *the value*

$$\widehat{\tau} = \arg\min_{\tau' \in \mathbb{R}} \mathop{\mathbb{P}}_{(\mathbf{x},y)\in S_{\text{train}}} [f^*(\mathbf{x}) \neq \textit{sign}(\widehat{\mathbf{v}} \cdot \mathbf{x} - \tau')].$$

satisfies

$$|\tau - \widehat{\tau}| \leq O\left(\frac{1}{\eta^{50}}\sqrt{\gamma}\right).$$

Formally, Proposition 35 follows from the two propositions above as follows. One first uses Proposition 38 to conclude that, for any absolute constant C_5, there is a value of the absolute constant C for which with probability $1 - \frac{1}{2000}$ a vector $\widehat{\mathbf{v}}$ that satisfies $\|\mathbf{v} - \widehat{\mathbf{v}}\| \leq \frac{1}{C_5}\beta^2\eta^{100}$. This implies that the angle between $\mathbf{v}$ and $\widehat{\mathbf{v}}$ is upper-bounded by $\frac{10}{C_5}\beta^2\eta^{100}$. Then, if the absolute constant C_5 is large enough, if we use Proposition 39, then with probability $1 - \frac{1}{2000}$ the value $\widehat{\tau}$ satisfies $|\tau - \widehat{\tau}| \leq \beta$, finishing the proof of Proposition 35.

Now, proceed to prove the two propositions above. We start with Proposition 38.

Proof of Proposition 38. Let $\{\mathbf{e}_1, \cdots \mathbf{e}_{d-1}\}$ form an orthonormal basis for the subspace orthogonal to $\mathbf{v}$. Since all the projections $\{\mathbf{v} \cdot \mathbf{x}, \mathbf{e}_1 \cdot \mathbf{x}, \cdots, \mathbf{e}_{d-1} \cdot \mathbf{x}\}$ are independent standard Gaussians and $f^*(\mathbf{x}) = \text{sign}(\mathbf{v} \cdot \mathbf{x} - \tau)$ we have for all i

$$\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)} [\mathbf{e}_i \cdot \mathbf{x} f^*(\mathbf{x})] = 0.$$

At the same time

$$\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)} [\mathbf{v} \cdot \mathbf{x} f(\mathbf{x})] = \int_{t=-\infty}^{+\infty} t\text{sign}(t-\tau)\frac{1}{\sqrt{2\pi}}\, dt =$$
$$\int_{t\in[-|\tau|,|\tau|]} t\text{sign}(t-\tau)\frac{1}{\sqrt{2\pi}}\, dt + \int_{t\in[-\infty,-|\tau|]\cup[|\tau|,+\infty]} t\text{sign}(t-\tau)\frac{1}{\sqrt{2\pi}}\, dt = \frac{2}{\sqrt{2\pi}}\int_{t=|\tau|}^{\infty} t\, dt$$

For some positive absolute constant K_2, the final expression above is at least $K_2\, \mathbb{P}_{t\sim N(0,1)}[t > \tau]$, because if $|\tau| > 1$, then one can lower-bound the expression above by $\frac{2}{\sqrt{2\pi}}\int_{t=|\tau|}^{\infty} dt$. On the other hand, if $|\tau| \in [0,1]$, then the expression on the right side is at least $\frac{2}{\sqrt{2\pi}}\int_{t=1}^{\infty} dt$ which is a positive absolute constant, while $\mathbb{P}_{t\sim N(0,1)}[t > \tau]$ is always upper-bounded by 1. Overall, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)} [\mathbf{v} \cdot \mathbf{x} f^*(\mathbf{x})] &\geq K_2 \mathop{\mathbb{P}}_{t\sim N(0,1)}[t > \tau] \\
&= K_2 \min\left(\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau > 0], \mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{v} \cdot \mathbf{x} - \tau < 0]\right) \\
&\geq K_2\eta.
\end{aligned}$$

Now, we bound the variance of $\mathbf{x}f^*(\mathbf{x})$. Since $f^*(\mathbf{x}) \in \{\pm 1\}$, we have

$$\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)} \left[(\mathbf{e}_i \cdot \mathbf{x} f^*(\mathbf{x}))^2\right] = \mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)} \left[(\mathbf{e}_i \cdot \mathbf{x})^2\right] = 1,$$

$$\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}\left[(\mathbf{v}\cdot\mathbf{x}f^*(\mathbf{x}))^2\right] = \mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}\left[(\mathbf{v}\cdot\mathbf{x})^2\right] = 1.$$

This allows us to use the Chebychev's inequality together with the union bound to conclude that with probability at least $1 - \frac{1}{2000}$ we have for all i

$$|E_{\mathbf{x}\in S}[\mathbf{e}_i \cdot \mathbf{x}f^*(\mathbf{x})]| \leq \sqrt{\frac{60d}{N}},$$

and also

$$E_{\mathbf{x}\in S}[\mathbf{v}\cdot\mathbf{x}f^*(\mathbf{x})] \geq K_2\eta - \sqrt{\frac{60d}{N}},$$

Recalling that $\widehat{\mathbf{v}} = \frac{\sum_{\mathbf{x}\in S_1}\mathbf{x}f^*(\mathbf{x})}{\left\|\sum_{\mathbf{x}\in S_1}\mathbf{x}f^*(\mathbf{x})\right\|_2} = \frac{\mathbb{E}_{\mathbf{x}\in S_1}\mathbf{x}f^*(\mathbf{x})}{\left\|\mathbb{E}_{\mathbf{x}\in S_1}\mathbf{x}f^*(\mathbf{x})\right\|_2}$, we see that

$$|\widehat{\mathbf{v}}\cdot\mathbf{e}_i| \leq \frac{\sqrt{\frac{60d}{N}}}{K_2\eta - \sqrt{\frac{60d}{N}}}$$

Substituting $N = C(\frac{d}{\eta\beta})^C$, and letting C be a sufficiently large absolute constant, we obtain from above implies that $|\widehat{\mathbf{v}}\cdot\mathbf{e}_i| \leq \frac{\beta}{10\sqrt{d}}$. Since $\|\widehat{\mathbf{v}}\| = 1$ we have

$$1 \geq |\widehat{\mathbf{v}}\cdot\mathbf{v}| \geq \sqrt{1-\frac{\beta}{10}} \geq 1 - \frac{\beta}{10},$$

we also see that taking C to be a sufficiently large absolute constant also ensures that $\widehat{\mathbf{v}}\cdot\mathbf{v} > 0$, so overall we get

$$\|\widehat{\mathbf{v}} - \mathbf{v}\| \leq \beta,$$

which finishes the proof. □

In order to prove Proposition 39, we will need a proposition that relates the following two quantities: (1) the difference in offsets τ_1 and τ_2 of two halfspaces (2) The probability that these two hafspaces disagree on a point drawn from the standard Gaussian.

Proposition 40. *There is some absolute constant K_1 such that for any pair of unit vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$ and a pair of real numbers τ_1, τ_2, letting γ denote the angle between $\mathbf{v}_1$ and $\mathbf{v}_2$, the following holds. Suppose $\gamma < \pi/4$, then*

$$\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}\left[\textit{sign}\left(\mathbf{v}_1\cdot\mathbf{x}-\tau_1\right) \neq \textit{sign}\left(\mathbf{v}_2\cdot\mathbf{x}-\tau_2\right)\right] \geq \frac{1}{K_1}e^{-\tau_1^2/2}\min\left(\left|\tau_1 - \frac{\tau_2}{\cos\gamma}\right|, \frac{1}{|\tau_1|+1}\right) \tag{3.17}$$

It is also the case that

$$\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}\left[\textit{sign}\left(\mathbf{v}_1\cdot\mathbf{x}-\tau_1\right) \neq \textit{sign}\left(\mathbf{v}_2\cdot\mathbf{x}-\tau_1\cos\gamma\right)\right] \leq K_1\sqrt{\gamma} \tag{3.18}$$

Proof. To prove this, we first show that for any $z \in \mathbb{R}$, conditioned on $\mathbf{v}_1 \cdot \mathbf{x} = z_1$ the distribution of $\mathbf{v}_2 \cdot \mathbf{x}$ is $\mathcal{N}(z_1 \cos\gamma, \sin\gamma)$. Indeed, let $\mathbf{v}_3$ be the unit vector that one obtains by first projecting $\mathbf{v}_2$ into the subspace perpendicular to $\mathbf{v}_1$, and then normalizing the resulting vector to have unit norm. This means $\mathbf{v}_3$ is orthogonal to $\mathbf{v}_1$ and we have

$$\mathbf{v}_2 = \mathbf{v}_1 \cos\gamma + \mathbf{v}_3 \sin\gamma.$$

Therefore

$$\mathbf{x} \cdot \mathbf{v}_2 = \mathbf{x} \cdot \mathbf{v}_1 \cos\gamma + \mathbf{x} \cdot \mathbf{v}_3 \sin\gamma$$

Now, since $\mathbf{x} \cdot \mathbf{v}_1$ and $\mathbf{x} \cdot \mathbf{v}_3$ are distributed as i.i.d. one-dimensional standard Gaussians. Thus, conditioning on $\mathbf{x} \cdot \mathbf{v}_1 = z_1$ we get that $\mathbf{x} \cdot \mathbf{v}_2$ is distributed as $\mathcal{N}(z\cos\gamma, \sin\gamma)$.

Our observation allows us to write:

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\text{sign}\,(\mathbf{v}_1\cdot\mathbf{x}-\tau_1) \neq \text{sign}\,(\mathbf{v}_2\cdot\mathbf{x}-\tau_2)] = \\
&\qquad \mathop{\mathbb{P}}_{z_1,z_2\in\mathcal{N}(0,1)}[\text{sign}\,(z_1-\tau_1) \neq \text{sign}\,(z_1\cos\gamma + z_2\sin\gamma - \tau_2)] = \\
&\qquad\qquad \mathop{\mathbb{P}}_{z_1,z_2\in\mathcal{N}(0,1)}[\text{sign}\,(z_1-\tau_1) \neq \text{sign}\,(z_1 + z_2\tan\gamma - \tau_2/\cos\gamma)] \quad (3.19)
\end{aligned}$$

Let us first focus on the case when $\gamma \in [0, \pi/2)$. We see that

$$\begin{aligned}
&\mathop{\mathbb{P}}_{z_1,z_2\in\mathcal{N}(0,1)}[\text{sign}\,(z_1-\tau_1) \neq \text{sign}\,(z_1 + z_2\tan\gamma - \tau_2/\cos\gamma)] \geq \\
&\qquad\qquad \frac{1}{2}\mathop{\mathbb{P}}_{z_1\in\mathcal{N}(0,1)}[\text{sign}\,(z_1-\tau_1) \neq \text{sign}\,(z_1 - \tau_2/\cos\gamma)] \quad (3.20)
\end{aligned}$$

The reason that inequality above is true is that, conditioned on a specific value of z_1, if $z_1 > \tau_2/\cos\gamma$, then $z_1 + z_2\tan\gamma - \tau_2$ is more likely to be positive than negative. At the same time, if $z_1 < \tau_2/\cos\gamma$, then $z_1 + z_2\tan\gamma - \tau_2$ is more likely to be negative than positive.

We lower-bound the probability above as follows. Let A be the interval of $\mathbb{R}$ defined as follows:

$$A := \left\{ z \in \mathbb{R} : \ \ \text{sign}(z-\tau_1) \neq \text{sign}(z - \tau_2/\cos\gamma) \ \ \&\ \ |z-\tau_1| \leq \frac{1}{|\tau_1|+1} \right\}$$

We have

$$\begin{aligned}
\mathop{\mathbb{P}}_{z_1\in\mathcal{N}(0,1)}[\text{sign}\,(z_1-\tau_1) \neq \text{sign}\,(z_1 - \tau_2/\cos\gamma)] &\geq \mathop{\mathbb{P}}_{z_1\in\mathcal{N}(0,1)}[z_1 \in A] \geq \\
&\geq \min\left(\left|\tau_1 - \frac{\tau_2}{\cos\gamma}\right|, \frac{1}{|\tau_1|+1}\right) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(|\tau_1| - \frac{1}{|\tau_1|+1}\right)^2} \\
&\geq \Omega(1) \cdot \min\left(\left|\tau_1 - \frac{\tau_2}{\cos\gamma}\right|, \frac{1}{|\tau_1|+1}\right) e^{-\tau_1^2/2}, \quad (3.21)
\end{aligned}$$

which, combined with Equations 3.19 and 3.20, finishes the proof of Equation 3.17.

Now, we proceed to proving Equation 3.18. We proceed as follows:

$$\begin{aligned}
&\mathop{\mathbb{P}}_{z_1,z_2\in\mathcal{N}(0,1)}\left[\text{sign}\left(z_1-\tau_1\right)=\text{sign}\left(z_1+z_2\tan\gamma-\tau_1\right)\right]\\
&\qquad\geq \mathop{\mathbb{P}}_{z_1,z_2\in\mathcal{N}(0,1)}\left[|z_1-\tau_1|>\sqrt{\tan\gamma}\ \ \&\ \ |z_2|<\frac{1}{\sqrt{\tan\gamma}}\right]\\
&\qquad\geq 1-O(1)\cdot\sqrt{\tan\gamma}-O(1)\int_{\frac{1}{\sqrt{\tan\gamma}}}^{\infty}e^{-z^2/2}\,dz\\
&\qquad= 1-O(\sqrt{\tan\gamma})=1-O(\sqrt{\gamma}),
\end{aligned}$$

which, when combining with with Equation 3.19 and substituting $\tau_2=\tau_1\cos\gamma$, proves Equation 3.18. □

Having proven Proposition 40, we are now ready to prove Proposition 39.

Proof of Proposition 39. Recall that $\mathcal{T}=\{\widehat{\mathbf{v}}\cdot\mathbf{x}:(\mathbf{x},y)\in S_{\text{train}}\}$. We see for τ' between two neighboring elements of $\mathcal{T}$ the value of $\mathbb{P}_{\mathbf{x}\in\mathbb{N}(0,I)}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau')]$ stays the same. Therefore

$$\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{T}}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\widehat{\tau})]=\min_{\tau'\in\mathcal{T}}\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{T}}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau')]=\min_{\tau'\in\mathbb{R}}\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{T}}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau')]. \tag{3.22}$$

Since the function class $\{\text{sign}(\mathbf{v}'\cdot\mathbf{x}-\tau':\ \mathbf{v}'\in\mathbb{R}^d,\ \tau'\in\mathbb{R}\}$ has a VC dimension of $d+1$, the standard VC bound tells us that for sufficiently large absolute constant C with probability at least $1-\frac{1}{2000}$ we have for every $\tau'\in\mathbb{R}$ and unit vector $\widehat{\mathbf{v}}$ that

$$\left|\mathop{\mathbb{P}}_{\mathbf{x}\in\mathbb{N}(0,I)}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau']-\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{T}}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau')]\right|\leq\sqrt{\gamma} \tag{3.23}$$

From Equation 3.18 in Proposition 40 we have that

$$\min_{\tau'\in\mathbb{R}}\mathop{\mathbb{P}}_{\mathbf{x}\in\mathbb{N}(0,I)}[f^*(\mathbf{x})\neq\text{sign}(\widehat{\mathbf{v}}\cdot\mathbf{x}-\tau')]\leq K_1\sqrt{\gamma}\leq O(\sqrt{\gamma}) \tag{3.24}$$

We now upper-bound $|\tau|$ in terms η as follows:

$$|\tau|\leq 10\sqrt{\log\frac{1}{\eta}}, \tag{3.25}$$

For $|\tau|<1$, this is immediate, because the probability that the Gaussian exceeds one standard deviation in a given direction is at least $1/10$. For $|\tau|\geq 1$, we write

$$\eta\geq\int_{|\tau|}^{\infty}e^{-t^2/2}\,dt\geq\frac{1}{|\tau|}e^{-(|\tau|+1/|\tau|)^2/2}\geq\frac{1}{e^2}\cdot\frac{1}{|\tau|}e^{-|\tau|^2/2},$$

which proves Equation 3.25.

Taking Equation 3.17 in Proposition 40 and substituting Equation 3.25 we get

$$\mathop{\mathbb{P}}_{x\in\mathcal{N}(0,I_d)}\left[f(x)\neq \operatorname{sign}\left(\widehat{\mathbf{v}}\cdot x-\widehat{\tau}\right)\right]\geq \frac{1}{K_1}e^{-\tau_1^2/2}\min\left(\left|\tau-\frac{\widehat{\tau}}{\cos\gamma}\right|,\frac{1}{|\tau|+1}\right)\geq \frac{\eta^{50}}{K_1}\min\left(\left|\tau-\frac{\widehat{\tau}}{\cos\gamma}\right|,1\right)$$

Combining the above with Equation 3.22, Equation 3.23 and Equation 3.24 we get

$$\left|\tau-\frac{\widehat{\tau}}{\cos\gamma}\right|\leq \frac{K_1}{\eta^{50}}(O(\sqrt{\gamma})+\sqrt{\gamma})\leq O(\sqrt{\gamma}/\eta^{50}).$$

Finally, we see that

$$|\tau-\widehat{\tau}|\leq\left|\tau-\frac{\widehat{\tau}}{\cos\gamma}\right|+\left|\widehat{\tau}-\frac{\widehat{\tau}}{\cos\gamma}\right|\leq O(\sqrt{\gamma}/\eta^{50})+O(\sqrt{\log(1/\eta)}\gamma^2)=O(\sqrt{\gamma}/\eta^{50}).$$

This completes the proof of Proposition 39. □

3.9 TDS Learning Through Moment Matching

3.9.1 $\mathcal{L}_2$-Sandwiching Implies TDS Learning

We now prove Theorem 16 which we restate here for convenience.

Theorem 22 ($\mathcal{L}_2$-sandwiching implies TDS Learning)**.** *Let D be a distribution over a set $\mathcal{X}\subseteq\mathbb{R}^d$ and let $\mathcal{C}\subseteq\{\mathcal{X}\to\{\pm1\}\}$ be a concept class. Let $\epsilon,\delta\in(0,1)$, $\epsilon'=\epsilon/100$ $\delta'=\delta/2$ and assume that the following are true.*

(*i*) *($\mathcal{L}_2$-Sandwiching) The ϵ'-approximate $\mathcal{L}_2$-sandwiching degree of $\mathcal{C}$ under D is at most k with coefficient bound B.*

(*ii*) *(Moment Concentration) If $X\sim D^{\otimes m}$ and $m\geq m_{\mathrm{conc}}$ then, with probability at least $1-\delta'$, we have that for any $\alpha\in\mathbb{N}^d$ with $\|\alpha\|_1\leq k$ it holds $|\mathbb{E}_X[\mathbf{x}^\alpha]-\mathbb{E}_D[\mathbf{x}^\alpha]|\leq\frac{\epsilon'}{B^2d^{4k}}$.*

(*iii*) *(Generalization) If $S\sim\mathcal{D}_{\mathcal{XY}}^{\otimes m}$ where $\mathcal{D}_{\mathcal{XY}}$ is any distribution over $\mathcal{X}\times\{\pm1\}$ such that $\mathcal{D}_{\mathcal{X}}=D$ and $m\geq m_{\mathrm{gen}}$ then, with probability at least $1-\delta'$ we have that for any degree-k polynomial p with coefficient bound B it holds $|\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2]-\mathbb{E}_S[(y-p(\mathbf{x}))^2]|\leq\epsilon'$.*

Then, Algorithm 5, upon receiving $m_{\mathrm{train}}\geq m_{\mathrm{gen}}$ labelled samples S_{train} from the training distribution and $m_{\mathrm{test}}\geq C\cdot\frac{d^k+\log(1/\delta)}{\epsilon^2}+m_{\mathrm{conc}}$ unlabelled samples X_{test} from the test distribution (where $C>0$ is a sufficiently large universal constant), runs in time $\mathrm{poly}(|S_{\mathrm{train}}|,|X_{\mathrm{test}}|,d^k)$ and TDS learns $\mathcal{C}$ with respect to D up to error $32\lambda+\epsilon$ and with failure probability δ.

Algorithm 5: TDS Learning through Moment Matching

Input: Sets S_{train} from $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$, X_{test} from $\mathcal{D}^{\text{test}}_{\mathcal{X}}$, parameters $\epsilon > 0, \delta \in (0,1)$, $k \in \mathbb{N}, B > 0$

Set $\epsilon' = \epsilon/100$, $\delta' = \delta/2$ and $\Delta = \frac{\epsilon'}{B^2 d^{4k}}$

For each $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq 2k$, compute the quantity

$\widehat{\mathrm{M}}_\alpha = \mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}^\alpha] = \mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[\prod_{i\in[d]} \mathbf{x}_i^{\alpha_i}\right]$

Reject and terminate if $|\widehat{\mathrm{M}}_\alpha - \mathbb{E}_{\mathbf{x}\sim D}[\mathbf{x}^\alpha]| > \Delta$ for some α with $\|\alpha\|_1 \leq 2k$.

Otherwise, solve the following least squares problem on S_{train} up to error ϵ'

$$\min_p \mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}\left[(y - p(\mathbf{x}))^2\right]$$
$$\text{s.t. } p \text{ is a polynomial with degree at most } k$$
$$\text{each coefficient of } p \text{ is absolutely bounded by } B$$

Let $\widehat{p}$ be an ϵ'-approximate solution to the above optimization problem.

Accept and output $h : \mathcal{X} \to \{\pm 1\}$ where $h : \mathbf{x} \mapsto \text{sign}(\widehat{p}(\mathbf{x}))$.

One key ingredient of the proof of Theorem 16 is the following transfer lemma which states that moment matching implies that the empirical squared loss between two polynomials on the test distribution is close to their expected squared loss under the target distribution.

Lemma 16 (Transfer Lemma for Square Loss)**.** *Let D be a distribution over $\mathcal{X} \subseteq \mathbb{R}^d$ and X_{test} a (multi)set of points in $\mathbb{R}^d$. If $|\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[\mathbf{x}^\alpha] - \mathbb{E}_{\mathbf{x}\sim D}[\mathbf{x}^\alpha]| \leq \Delta$ for all $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq 2k$, then for any degree k polynomials p_1, p_2 with coefficients that are absolutely bounded by B, it holds*

$$\left|\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2] - \mathbb{E}_{\mathbf{x}\sim D}[(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2]\right| \leq B^2 \cdot d^{4k} \cdot \Delta$$

Proof. The polynomials p_1, p_2 all have degree at most k and coefficients that are absolutely bounded by B. Therefore, the polynomial $(p_1 - p_2)^2$ has degree at most $2k$ and coefficients that are absolutely bounded by $B^2 d^{2k}$. Let $p' = (p_1 - p_2)^2 = \sum_{\alpha:\|\alpha\|_1 \leq 2k} p'_\alpha \mathbf{x}^\alpha$ (with $|p'_\alpha| \leq B^2 d^{2k}$ as argued above) which gives the following.

$$\|p_1 - p_2\|^2_{\mathcal{L}_2(X_{\text{test}})} = \mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2\right] = \mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[p'(\mathbf{x})\right]$$

It remains to relate $\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}[p'(\mathbf{x})]$ to $\mathbb{E}_{\mathbf{x}\sim D}[p'(\mathbf{x})]$, which follows by the moment-matching as-

sumption.

$$\begin{aligned}\left|\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[p'(\mathbf{x})\right]-\mathbb{E}_{\mathbf{x}\sim D}\left[p'(\mathbf{x})\right]\right| &= \left|\sum_{\alpha:\|\alpha\|_1\le 2k} p'_\alpha\left(\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[\mathbf{x}^\alpha\right]-\mathbb{E}_{\mathbf{x}\sim D}\left[\mathbf{x}^\alpha\right]\right)\right| \\ &\le \sum_{\alpha:\|\alpha\|_1\le 2k} |p'_\alpha|\cdot\left|\mathbb{E}_{\mathbf{x}\sim X_{\text{test}}}\left[\mathbf{x}^\alpha\right]-\mathbb{E}_{\mathbf{x}\sim D}\left[\mathbf{x}^\alpha\right]\right| \\ &= \sum_{\alpha:\|\alpha\|_1\le 2k} |p'_\alpha|\cdot\left|\widehat{\mathrm{M}}_\alpha-\mathrm{M}_\alpha\right| \\ &\le d^{2k}\cdot B^2\cdot d^{2k}\cdot\Delta\,,\end{aligned}$$

which concludes the proof of the lemma. □

We are now ready to prove Theorem 16.

Proof of Theorem 16. For the following, let $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ be the training distribution, $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$ the test distribution (both over $\mathcal{X}\times\{\pm 1\}$) and $\mathcal{D}^{\text{train}}_{\mathcal{X}}, \mathcal{D}^{\text{test}}_{\mathcal{X}}$ the corresponding marginal distributions over $\mathcal{X}$. We assume that $\mathcal{D}^{\text{train}}_{\mathcal{X}} = D$. Let $m_{\text{train}} = |S_{\text{train}}|$ and $m_{\text{test}} = |X_{\text{test}}|$, $\epsilon' = \epsilon/100$, $\delta' = \delta/2$, k, B as defined in condition (i). We also set $\Delta = \frac{\epsilon'}{B^2 d^{4k}}$ and m_{conc} as defined in condition (ii), as well as m_{gen} as defined in (iii).

Soundness. Suppose that Algorithm 5 accepts and outputs $h = \text{sign}(\widehat{p})$. For the following, let $\lambda_{\text{train}} = \text{err}(f^*; \mathcal{D}^{\text{train}}_{\mathcal{XY}})$ and $\lambda_{\text{test}} = \text{err}(f^*; \mathcal{D}^{\text{test}}_{\mathcal{XY}})$ (where we have $\lambda = \lambda_{\text{train}} + \lambda_{\text{test}}$). We can bound the error of the hypothesis h on $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$ as follows

$$\begin{aligned}\text{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) &\le \text{err}(f^*; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) + \text{err}(f^*, h; \mathcal{D}^{\text{test}}_{\mathcal{X}}) \\ &= \lambda_{\text{test}} + \mathbb{E}[\text{err}(f^*, h; X_{\text{test}})]\,,\end{aligned}$$

where the expectation above is over $X_{\text{test}} \sim (\mathcal{D}^{\text{test}}_{\mathcal{X}})^{\otimes m_{\text{test}}}$. Denote $\text{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) = \mathbb{P}_{\mathcal{D}^{\text{test}}_{\mathcal{XY}}}[y \neq h(\mathbf{x})]$ and $\text{err}(h_1, h_2; \mathcal{D}^{\text{test}}_{\mathcal{X}}) = \mathbb{P}_{\mathcal{D}^{\text{test}}_{\mathcal{X}}}[h_1(\mathbf{x}) \neq h_2(\mathbf{x})]$ and use the fact that for random variables $y_1, y_2, y_3 \in \{\pm 1\}$, it holds $\mathbb{P}[y_1 \neq y_2] \le \mathbb{P}[y_1 \neq y_3] + \mathbb{P}[y_2 \neq y_3]$. Since h is the sign of a polynomial with degree at most $k = k(\epsilon')$ (see Algorithm 5) and the class of functions of this form has VC dimension at most d^k (e.g., by viewing it as the class of halfspaces in d^k dimensions) we have that whenever $m_{\text{test}} \ge C\cdot\frac{d^k + \log(1/\delta')}{\epsilon'^2}$ for some sufficiently large universal constant $C > 0$ the following is true with probability at least $1-\delta'$ over the distribution of X_{test}.

$$\mathbb{E}[\text{err}(f^*, h; X_{\text{test}})] \le \text{err}(f^*, h; X_{\text{test}}) + \epsilon'$$

Therefore, it is sufficient to bound the quantity $\mathrm{err}(f^*, h; X_{\mathrm{test}})$. We now observe the following simple fact.

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim X_{\mathrm{test}}}[(f^*(\mathbf{x}) - \widehat{p}(\mathbf{x}))^2] &\geq \mathop{\mathbb{P}}_{X_{\mathrm{test}}}[f^*(\mathbf{x}) = 1, \widehat{p}(\mathbf{x}) < 0] + \mathop{\mathbb{P}}_{X_{\mathrm{test}}}[f^*(\mathbf{x}) = -1, \widehat{p}(\mathbf{x}) \geq 0] \\ &= \mathop{\mathbb{P}}_{X_{\mathrm{test}}}[f^*(\mathbf{x}) \neq \mathrm{sign}\widehat{p}(\mathbf{x})] \\ &= \mathrm{err}(f^*, h; X_{\mathrm{test}})\end{aligned}$$

Therefore, we have $\mathrm{err}(f^*, h; X_{\mathrm{test}}) \leq \|f^* - \widehat{p}\|^2_{\mathcal{L}_2(X_{\mathrm{test}})}$. Let $p_{\mathrm{up}}, p_{\mathrm{down}}$ be ϵ'-approximate $\mathcal{L}_2$ sandwiching polynomials for f^* of degree at most $k = k(\epsilon')$ and with coefficient bound $B = B(\epsilon')$. The right hand side can be bounded as follows.

$$\begin{aligned}\|f^* - \widehat{p}\|_{\mathcal{L}_2(X_{\mathrm{test}})} &\leq \|f^* - p_{\mathrm{down}}\|_{\mathcal{L}_2(X_{\mathrm{test}})} + \|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(X_{\mathrm{test}})} \\ &\leq \|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(X_{\mathrm{test}})} + \|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(X_{\mathrm{test}})}\end{aligned}$$

In the last inequality, we used the fact that $p_{\mathrm{down}}(\mathbf{x}) \leq f^*(\mathbf{x}) \leq p_{\mathrm{up}}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$. We will now compare $\|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(X_{\mathrm{test}})}$ to $\|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(D)}$ (and, similarly, $\|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(X_{\mathrm{test}})}$ to $\|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(D)}$) using the transfer lemma (Lemma 16). The polynomials $p_{\mathrm{up}}, p_{\mathrm{down}}, \widehat{p}$ all have degree at most k and coefficients that are absolutely bounded by B. Moreover, since Algorithm 5 has accepted, we have that for any $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq 2k$, the following is true

$$\left|\widehat{\mathrm{M}}_\alpha - \mathrm{M}_\alpha\right| \leq \Delta\,, \tag{3.26}$$

where $\widehat{\mathrm{M}} = \mathbb{E}_{\mathbf{x}\sim X_{\mathrm{test}}}[\mathbf{x}^\alpha]$ (recall that $\mathbf{x}^\alpha = \prod_{i\in[d]} \mathbf{x}_i^{\alpha_i}$), $\mathrm{M} = \mathbb{E}_{\mathbf{x}\sim D}[\mathbf{x}^\alpha]$ and $\Delta = \frac{\epsilon'}{B^2 d^{4k}}$. Therefore, by applying Lemma 16, we obtain that $\|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(X_{\mathrm{test}})} \leq \|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(D)} + \sqrt{\epsilon'}$ and, similarly, $\|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(X_{\mathrm{test}})} \leq \|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(D)} + \sqrt{\epsilon'}$.

We have assumed that $p_{\mathrm{up}}, p_{\mathrm{down}}$ are ϵ'-approximate $\mathcal{L}_2$ sandwiching polynomials for f^* and, therefore $\|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(D)} = \sqrt{\|p_{\mathrm{up}} - p_{\mathrm{down}}\|^2_{\mathcal{L}_2(D)}} \leq \sqrt{\epsilon'}$ (see Definition 11). We bound the quantity $\|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(D)}$ as follows.

$$\begin{aligned}\|p_{\mathrm{down}} - \widehat{p}\|_{\mathcal{L}_2(D)} &\leq \|p_{\mathrm{down}} - f^*\|_{\mathcal{L}_2(D)} + \|f^* - \widehat{p}\|_{\mathcal{L}_2(D)} \\ &\leq \|p_{\mathrm{up}} - p_{\mathrm{down}}\|_{\mathcal{L}_2(D)} + \|f^* - \widehat{p}\|_{\mathcal{L}_2(D)} \qquad \text{(since } p_{\mathrm{down}} \leq f^* \leq p_{\mathrm{up}}\text{)} \\ &\leq \sqrt{\epsilon'} + \|f^* - \widehat{p}\|_{\mathcal{L}_2(D)} \qquad (3.27)\end{aligned}$$

Recall that $\|f^* - \widehat{p}\|^2_{\mathcal{L}_2(D)} = \mathbb{E}_{\mathbf{x}\sim D}[(\widehat{p}(\mathbf{x}) - f^*(\mathbf{x}))^2]$. By assumption, $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}} = D$ and therefore $\mathbb{E}_{\mathbf{x}\sim D}[(\widehat{p}(\mathbf{x}) - f^*(\mathbf{x}))^2] = \mathbb{E}_{\mathbf{x}\sim \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}}[(\widehat{p}(\mathbf{x}) - f^*(\mathbf{x}))^2]$. Moreover, we can view the expectation to be over the joint distribution $(\mathbf{x}, y) \sim \mathcal{D}^{\mathrm{train}}_{\mathcal{X}\mathcal{Y}}$ (coupling of $\mathbf{x}$ and y), but the variable y is ignored, i.e., $\mathbb{E}_{\mathbf{x}\sim \mathcal{D}^{\mathrm{train}}_{\mathcal{X}}}[(\widehat{p}(\mathbf{x}) - f^*(\mathbf{x}))^2] = \mathbb{E}_{(\mathbf{x},y)\sim \mathcal{D}^{\mathrm{train}}_{\mathcal{X}\mathcal{Y}}}[(\widehat{p}(\mathbf{x}) - f^*(\mathbf{x}))^2]$. We can bound the latter term as

follows.

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-f^*(\mathbf{x}))^2]^{1/2} &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-y+y-f^*(\mathbf{x}))^2]^{1/2}\\ &\leq \mathbb{E}_{\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-y)^2]^{1/2}+\mathbb{E}_{\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]^{1/2}\end{aligned}$$

For the term $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-y)^2]$, we use condition (iii) to have with probability at least $1-\delta'$, $|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-y)^2]-\mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}[(\widehat{p}(\mathbf{x})-y)^2]|\leq\epsilon'$ whenever $m_{\text{train}}\geq m_{\text{gen}}$. We now use the fact that $\widehat{p}$ is an ϵ'-approximate solution to the least squares problem defined in Algorithm 5 and have the following bound

$$\mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}[(\widehat{p}(\mathbf{x})-y)^2]^{1/2}\leq\mathbb{E}_{(\mathbf{x},y)\sim S_{\text{train}}}[(p_{\text{down}}(\mathbf{x})-y)^2]^{1/2}+\sqrt{\epsilon'}$$

Therefore, due to the generalization condition we have

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(\widehat{p}(\mathbf{x})-y)^2]^{1/2} &\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(p_{\text{down}}(\mathbf{x})-y)^2]^{1/2}+3\sqrt{\epsilon'}\\ &\leq \|p_{\text{down}}-f^*\|_{\mathcal{L}_2(\mathcal{D}^{\text{train}}_{\mathcal{X}})}+\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]^{1/2}+3\sqrt{\epsilon'}\\ &\leq \|p_{\text{down}}-p_{\text{up}}\|_{\mathcal{L}_2(D)}+\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]^{1/2}+3\sqrt{\epsilon'}\\ &\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]^{1/2}+4\sqrt{\epsilon'}\end{aligned}$$

Therefore, we have shown that $\|f^*-\widehat{p}\|_{\mathcal{L}_2(D)}\leq 4\mathbb{E}_{\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]^{1/2}+2\sqrt{\epsilon'}$. Note that $\mathbb{E}_{\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[(y-f^*(\mathbf{x}))^2]=4\,\mathbb{P}_{\mathcal{D}^{\text{train}}_{\mathcal{XY}}}[y\neq f^*(\mathbf{x})]=4\lambda_{\text{train}}$. Therefore, $\|f^*-\widehat{p}\|_{\mathcal{L}_2(D)}\leq 4\sqrt{\lambda_{\text{train}}}+4\sqrt{\epsilon'}$. By Equation (3.27), this implies $\|p_{\text{down}}-\widehat{p}\|_{\mathcal{L}_2(D)}\leq 4\sqrt{\lambda_{\text{train}}}+5\sqrt{\epsilon'}$, which in turn implies $\|p_{\text{down}}-\widehat{p}\|_{\mathcal{L}_2(X_{\text{test}})}\leq 4\sqrt{\lambda_{\text{train}}}+7\sqrt{\epsilon'}$. We overall obtain the following bound.

$$\begin{aligned}\text{err}(h;\mathcal{D}^{\text{test}}_{\mathcal{XY}}) &\leq \lambda_{\text{test}}+(4\lambda^{1/2}_{\text{train}}+7\sqrt{\epsilon'})^2\\ &\leq \lambda_{\text{test}}+32\lambda_{\text{train}}+100\epsilon'\\ &\leq 32\lambda+\epsilon \qquad\qquad \text{(since } \epsilon'=\epsilon/100 \text{ and } \lambda_{\text{test}}\geq 0)\end{aligned}$$

Note that, in fact, we have also demonstrated that upon acceptance, the following is true.

$$\text{err}(f^*,h;\mathcal{D}^{\text{test}}_{\mathcal{X}})\leq 32\lambda_{\text{train}}+\epsilon$$

The results above holds with probability at least $1-3\delta'=1-\delta$ (union bound over two bad events).

Completeness. For completeness, it is sufficient to ensure that $m_{\text{test}}\geq m_{\text{conc}}$, because then, the probability of acceptance is at least $1-\delta$, due to condition (ii), as required. □

3.9.2 Applications

In this section, we apply our main result in Theorem 16 to obtain a number of TDS learners for important concept classes with respect to Gaussian and Uniform target marginals. In particular, we will use the following corollary, which follows by Theorem 16 and some simple properties of the Gaussian and Uniform distributions (see Lemmas 20 and 21).

Corollary 1. *Let D be either the standard Gaussian in d dimensions or the uniform distribution over the d-dimensional hypercube. Let $\mathcal{C}$ be a concept class whose ϵ-approximate sandwiching degree with respect to D is k. Then, there is an algorithm that runs in time $d^{O(k)}$ and TDS learns $\mathcal{C}$ up to error $32\lambda + O(\epsilon)$ and failure probability at most 0.1.*

Boolean Classes. We now bound the $\mathcal{L}_2$ sandwiching degree of bounded size Decision trees and bounded size and depth Boolean Formulas.

Lemma 17 ($\mathcal{L}_2$ sandwiching degree of Decision Trees)**.** *Let D be the uniform distribution over the hypercube $\mathcal{X} = \{\pm 1\}^d$. For $s \in \mathbb{N}$, let $\mathcal{C}$ be the class of Decision Trees of size s. Then, for any $\epsilon > 0$ the $\mathcal{L}_2$ sandwiching degree of $\mathcal{C}$ is at most $k = O(\log(s/\epsilon))$.*

Proof. Let $f \in \mathcal{C}$ be a decision tree of size s. Consider the polynomials $p_{\mathrm{up}}, p_{\mathrm{down}}$ over $\{\pm 1\}^d$ which correspond to the following truncated decision trees. For p_{up}, we truncate f at depth k and substitute the internal nodes at depth k with leaf nodes labelled 1. Then, p_{up} corresponds to a sum of polynomials of degree at most k, each corresponding to a root-to-leaf path in the truncated decision tree. Clearly, $p_{\mathrm{up}} \geq f$ and p_{up} has degree k. We have that $\mathbb{E}_D[(p_{\mathrm{up}}(\mathbf{x}) - f(\mathbf{x}))^2]$ is upper bounded by a constant multiple of the probability that p_{up} takes the value 1, while $f(\mathbf{x})$ takes the value -1, since p_{up} is itself a Boolean-valued function (it is a decision tree). The probability that this happens is at most $s \cdot 2^{-k} = O(\epsilon)$ for $k = O(\log(s/\epsilon))$. We obtain p_{down} by a symmetric argument. □

For the following lemma, we make use of an upper bound for the pointwise distance between a Boolean formula and the best approximating low-degree polynomial from [OS03] (which readily implies the existence of low-degree $\mathcal{L}_2$ sandwiching polynomials).

Lemma 18 ($\mathcal{L}_2$ SD of Boolean Formulas, Theorem 6 in [OS03])**.** *Let D be the uniform distribution over the hypercube $\mathcal{X} = \{\pm 1\}^d$. For $s, \ell \in \mathbb{N}$, let $\mathcal{C}$ be the class of Boolean formulas of size at most s, depth at most ℓ. Then, for any $\epsilon > 0$ the $\mathcal{L}_2$ sandwiching degree of $\mathcal{C}$ is at most $k = (C \log(s/\epsilon))^{5\ell/2}\sqrt{s}$, for some sufficiently large universal constant $C > 0$.*

Proof. Let $f \in \mathcal{C}$ be an formula of size s and depth ℓ. We first construct a polynomial p that satisfies $|p(\mathbf{x}) - f(\mathbf{x})| \leq \sqrt{\epsilon}/2$ for any $\mathbf{x} \in \{\pm 1\}^d$. This corresponds to a slight modification of the proof of Theorem 6 in [OS03], where the basis of the inductive construction of p (see Lemma 10 in [OS03]) is an $O(\sqrt{\epsilon}/s^3)$ bound (instead of the original $1/s^3$ bound) for the (trivial) approximation of a single variable $\mathbf{x}_i$ by itself. The degree of p is indeed upper bounded by $(C \log(s/\epsilon))^{5\ell/2}\sqrt{s}$ and we may obtain $p_{\mathrm{up}}, p_{\mathrm{down}}$ by setting $p_{\mathrm{up}}(\mathbf{x}) = p(\mathbf{x}) + \sqrt{\epsilon}/2$ and $p_{\mathrm{down}} = p(\mathbf{x}) - \sqrt{\epsilon}/2$.

Clearly, $p_{\mathrm{down}}(\mathbf{x}) \leq f(\mathbf{x}) \leq p_{\mathrm{up}}(\mathbf{x})$ and $|p_{\mathrm{up}}(\mathbf{x}) - p_{\mathrm{down}}(\mathbf{x})| = \sqrt{\epsilon}$ for all $\mathbf{x} \in \{\pm 1\}^d$. Therefore $\|p_{\mathrm{up}} - p_{\mathrm{down}}\|^2_{\mathcal{L}_2(D)} \leq \epsilon$. □

We obtain the following results for agnostic TDS learning of boolean concept classes.

Corollary 2 (TDS Learner for Decision Trees)**.** *Let D be the uniform distribution over the hypercube in d dimensions. Then, there is an algorithm that runs in time $d^{O(\log(s/\epsilon))}$ and TDS learns Decision Trees of size s with respect to $\mathrm{Unif}(\{\pm 1\}^d)$ up to error $32\lambda + O(\epsilon)$.*

Corollary 3 (TDS Learner for Boolean Formulas)**.** *Let D be the uniform distribution over the hypercube in d dimensions and $C > 0$ some sufficiently large universal constant. Then, there is an algorithm that runs in time $d^{\sqrt{s}(C\log(s/\epsilon))^{5\ell/2}}$ and TDS learns Boolean formulas of size at most s and depth at most ℓ with respect to $\mathrm{Unif}(\{\pm 1\}^d)$ up to error $32\lambda + O(\epsilon)$.*

Intersections and Decision Trees of Halfspaces. We now provide an upper bound for the $\mathcal{L}_2$-sandwiching degree of Decision Trees of halfspaces, which does not merely follow from a bound on the $\mathcal{L}_\infty$ approximate degree and, in particular, holds under both the Gaussian distribution and the Uniform over the hypercube. The following lemma is based on a powerful result from pseudorandomness literature (Theorem 10.4 from [GOWZ10]) which was originally used to provide a bound for the $\mathcal{L}_1$-sandwiching degree of decision trees of halfspaces, but, as we show, also provides a bound on the $\mathcal{L}_2$-sandwiching degree with careful manipulation.

Lemma 19 ($\mathcal{L}_2$-sandwiching degree of Intersections and Decision Trees of Halfspaces)**.** *Let D be either the uniform distribution over the hypercube $\mathcal{X} = \{\pm 1\}^d$ or the multivariate Gaussian distribution $\mathcal{N}(0, I_d)$ over $\mathcal{X} = \mathbb{R}^d$. For $\ell \in \mathbb{N}$, let also $\mathcal{C}$ be the class of concepts that can be expressed as an intersection of ℓ halfspaces on $\mathcal{X}$. Then, for any $\epsilon > 0$ the $\mathcal{L}_2$ sandwiching degree of $\mathcal{C}$ is at most $k = \widetilde{O}(\frac{\ell^6}{\epsilon^2})$. For Decision Trees of halfspaces of size s and depth ℓ, the bound is $k = \widetilde{O}(\frac{s^2\ell^6}{\epsilon^2})$.*

The above result implies the following corollary.

Corollary 4 (TDS Learner for Intersections and Decision Trees of Halfspaces)**.** *Let D be either the standard Gaussian in $\mathbb{R}^d$ or the uniform distribution over the hypercube in d dimensions. Then, there is an algorithm that runs in time $d^{\tilde{O}(\ell^6/\epsilon^2)}$ and TDS learns intersections of ℓ halfspaces with respect to D up to error $32\lambda + O(\epsilon)$. For Decision Trees of halfspaces with size s and depth ℓ the bound is $d^{\tilde{O}(s^2\ell^6/\epsilon^2)}$.*

In order to apply the structural result we need from [GOWZ10], we first provide a formal definition for the notion of hypercontractivity.

Definition 13 (Hypercontractivity)**.** *Let D_1 be a distribution over $\mathbb{R}$ and let $T \in \mathbb{N}$, $T > 2$, $\eta \in (0, 1)$. We say that D_1 is $(T, 2, \eta)$-hypercontractive if $\mathbb{E}[x^T] < \infty$ and for any $a \in \mathbb{R}$ we have*

$$\mathbb{E}_{x\sim D_1}[(a + \eta x)^T]^{1/T} \leq \mathbb{E}_{x\sim D_1}[(a + \eta x)^2]^{1/2}$$

The following result can be used to show Lemma 19.

Proposition 41 (Modification of Theorem 10.4 from [GOWZ10])**.** *Let $r \in \mathbb{N}$, $\sigma \in (0,1)$, $T \in \mathbb{N}$, $\eta > 0$ and $t > 4$ be parameters and consider D to be a product distribution over $\mathcal{X} \subseteq \mathbb{R}^d$ such that each of its independent coordinates is $(4, 2, \eta)$-hypercontractive, and $(T, 2, 4/t)$-hypercontractive. Suppose that $T \geq Cr\log(rt)$ for some sufficiently large universal constant $C > 0$ and T is even. Then, for any function of the form $h : \mathcal{X} \to \mathbb{R}$, $h(\mathbf{x}) = \mathbb{1}\{\mathbf{w} \cdot \mathbf{x} \geq \tau\}$, where $\mathbf{w} \in \mathbb{R}^d$ and $\tau \in \mathbb{R}$, there is a polynomial $p : \mathcal{X} \to \mathbb{R}$ such that the following are true.*

(*i*) *The degree of p is at most $k = \mathrm{poly}(\log t, \frac{1}{\eta}) \cdot \frac{1}{\sigma} + O(\frac{T}{r})$.*

(*ii*) *For any $\mathbf{x} \in \mathcal{X}$ we have $p(\mathbf{x}) \geq h(\mathbf{x})$.*

(*iii*) *The expected distance between p and h is bounded by $\mathbb{E}_{\mathbf{x}\sim D}[p(\mathbf{x}) - h(\mathbf{x})] \leq O(\sigma^{\frac{1}{2}} + \frac{rt\log(rt)}{T})$.*

(*iv*) *The values of p are upper bounded with high probability, i.e., $\mathbb{P}_{\mathbf{x}\sim D}[p(\mathbf{x}) > 1 + \frac{1}{r^2}] \leq 2^{-T/r}$.*

(*v*) *The $L_{2r}(D)$ norm of p is bounded by $\|p\|_{L_{2r}(D)} \leq 1 + \frac{2}{r^2}$.*

Proof of Lemma 19. Let $f \in \mathcal{C}$ be an intersection of ℓ halfspaces over $\mathcal{X}$, i.e., f can be written in the following form

$$f(\mathbf{x}) = 2\prod_{j=1}^{\ell} h_j(\mathbf{x}) - 1, \text{ where } h_j(\mathbf{x}) = \mathbb{1}\{\mathbf{w}_j \cdot \mathbf{x} + \tau_j\} \text{ for some } \mathbf{w}_j \in \mathbb{R}^d, \tau_j \in \mathbb{R}$$

Note that if f is a Decision Tree of halfspaces of size s and depth ℓ, then f can be written as a sum of at most s intersections of ℓ halfspaces and it suffices to use accuracy parameter ϵ/s for each intersection.

Back to the case where f is an intersection of ℓ halfspaces, we will apply Proposition 41 in a way similar to the proof of Lemma 10.1 in [GOWZ10]. However, our goal here is to show that Proposition 41 implies the existence of $\mathcal{L}_2$ (rather than $\mathcal{L}_1$) sandwiching polynomials for f. We use the following standard fact about the Gaussian and Uniform distributions.

Claim 5. *(*Hypercontractivity of Gaussian and Uniform marginals, see e.g. [KS88, Wol07, GOWZ10]*) If D is either the standard Gaussian $\mathcal{N}(0, I_d)$ over $\mathbb{R}^d$ or the uniform distribution over the hypercube $\{\pm 1\}^d$, then, for some universal constant $C > 0$, each of the coordinates of D is $(\lceil Ct^2 \rceil, 2, \frac{4}{t})$-hypercontractive for any $t > 0$ and, in particular, each one is also $(4, 2, \frac{1}{\sqrt{3}})$-hypercontractive.*

We may apply Proposition 41 for each h_j with parameters $r = 2\ell, \sigma = \frac{\epsilon^2}{C\ell^4}, t = C\frac{\ell^3}{\epsilon}\log(\ell/\epsilon)$, $\eta = 1/\sqrt{3}$ and $T = Ct^2$, for some sufficiently large universal constant C to obtain a polynomial

p_j of degree $k = \widetilde{O}(\frac{\ell^5}{\epsilon^2})$ such that the following are true.

$$p_j(\mathbf{x}) \geq h_j(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathcal{X} \tag{3.28}$$

$$\epsilon_1 := \mathbb{E}_D[p_j(\mathbf{x}) - h_j(\mathbf{x})] = O\Big(\frac{\epsilon}{\ell^2}\Big) \tag{3.29}$$

$$\epsilon_2 := \mathop{\mathbb{P}}_D\Big[p_j(\mathbf{x}) > 1 + \frac{1}{4\ell^2}\Big] \leq 2^{-\Omega(\frac{\ell^5}{\epsilon^2}\log^2(\ell/\epsilon))} \tag{3.30}$$

$$\|p_j\|_{L_{4m}(D)} \leq 1 + \frac{1}{2\ell^2} \tag{3.31}$$

We will now construct a polynomial p_{up} of degree $\widetilde{O}(\frac{\ell^6}{\epsilon^2})$ such that $p_{\text{up}}(\mathbf{x}) \geq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$ and also $\mathbb{E}_D[(p_{\text{up}}(\mathbf{x}) - f(\mathbf{x}))^2] \leq \epsilon/4$. This implies the existence of a corresponding polynomial p_{down} with $p_{\text{down}}(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbb{E}_D[(p_{\text{up}}(\mathbf{x}) - p_{\text{down}}(\mathbf{x}))^2] \leq \epsilon$ via a symmetric argument. Our proof consists of a hybrid argument similar to the one used in the proof of Lemma 10.1 in [GOWZ10], modified to provide a bound for the $\mathcal{L}_2$ error of approximation.

We pick $p_{\text{up}} = 2p - 1$, where $p = \prod_{j=1}^{\ell} p_j$. Let $p^{(0)} = \prod_{j=1}^{\ell} h_j$, $p^{(i)} = (\prod_{j=1}^{i} p_j)(\prod_{j=i+1}^{\ell} h_j)$ and $p^{(\ell)} = p$. We then have the following.

$$\begin{aligned}
\|p - h\|_{\mathcal{L}_2(D)} = \|p^{(\ell)} - p^{(0)}\|_{\mathcal{L}_2(D)} &\leq \sum_{i=1}^{\ell} \|p^{(i)} - p^{(i-1)}\|_{\mathcal{L}_2(D)} \\
&= \sum_{i=1}^{\ell} \Big\| \Big(\prod_{j=1}^{i-1} p_j\Big)\Big(\prod_{j=i+1}^{\ell} h_j\Big)(p_i - h_i)\Big\|_{\mathcal{L}_2(D)} \\
&\leq \sum_{i=1}^{\ell} \Big\| \Big(\prod_{j\neq i} p_j\Big)(p_i - h_i)\Big\|_{\mathcal{L}_2(D)} \qquad \text{(by property (3.28))}
\end{aligned}$$

For any fixed $i \in [\ell]$ we have

$$\begin{aligned}
\Big\| \Big(\prod_{j\neq i} p_j\Big)(p_i - h_i)\Big\|^2_{\mathcal{L}_2(D)} &= \mathbb{E}_D\Big[\Big(\prod_{j\neq i} p_j^2(\mathbf{x})\Big)(p_i(\mathbf{x}) - h_i(\mathbf{x}))^2\Big] \\
&\leq \mathbb{E}_D\Big[\Big(\prod_{j\neq i} p_j^2(\mathbf{x})\Big)(p_i(\mathbf{x}) - h_i(\mathbf{x}))p_i(\mathbf{x})\Big] \quad \text{(since } h_i \geq 0 \text{ and } p_i \geq h_i\text{)}
\end{aligned}$$

In order to bound the quantity $\mathbb{E}_D[(\prod_{j\neq i} p_j^2)(p_i - h_i)p_i]$, we split the expectation according to the event $\mathcal{E}$ that $(\prod_{j\neq i} p_j)\sqrt{p_i} < 2$. In particular, we have that $\mathbb{E}_D[(\prod_{j\neq i} p_j^2)(p_i - h_i)p_i \, \mathbb{1}\{\mathcal{E}\}]$ is at

most $4\epsilon_1$ by property (3.29) and $\mathbb{E}_D[(\prod_{j\neq i} p_j^2)(p_i - h_i)p_i \mathbb{1}\{\neg\mathcal{E}\}]$ is bounded as follows.

$$\mathbb{E}_D\Big[\Big(\prod_{j\neq i} p_j^2(\mathbf{x})\Big)(p_i(\mathbf{x}) - h_i(\mathbf{x}))p_i(\mathbf{x})\,\mathbb{1}\Big\{\Big(\prod_{j\neq i} p_j(\mathbf{x})\Big)\sqrt{p_i(\mathbf{x})} \geq 2\Big\}\Big] \leq$$
$$\leq \mathbb{E}_D\Big[\Big(\prod_{j\in[\ell]} p_j^2(\mathbf{x})\Big)\,\mathbb{1}\Big\{\Big(\prod_{j\neq i} p_j(\mathbf{x})\Big)\sqrt{p_i(\mathbf{x})} \geq 2\Big\}\Big] \qquad \text{(by property (3.28))}$$

We now observe that whenever $(\prod_{j\neq i} p_j(\mathbf{x}))\sqrt{p_i(\mathbf{x})} \geq 2$, there must exist some index j' such that $p_{j'}(\mathbf{x}) > 1 + \frac{1}{4\ell^2}$ and, therefore, we can further bound the above quantity by the following one.

$$\mathbb{E}_D\Big[\sum_{j'=1}^{\ell} \mathbb{1}\Big\{p_{j'}(\mathbf{x}) > 1 + \frac{1}{4\ell^2}\Big\}\Big(\prod_{j\in[\ell]} p_j^2(\mathbf{x})\Big)\Big] = \sum_{j'=1}^{\ell} \mathbb{E}_D\Big[\,\mathbb{1}\Big\{p_{j'}(\mathbf{x}) > 1 + \frac{1}{4\ell^2}\Big\}\Big(\prod_{j\in[\ell]} p_j^2(\mathbf{x})\Big)\Big]$$

In the above expression we used linearity of expectation. We now apply Hölder's inequality and obtain the bound $\sum_{j'=1}^{\ell}(\mathbb{P}_D[p_{j'}(\mathbf{x}) > 1 + \frac{1}{4\ell^2}])^{\frac{1}{2}} \prod_{j=1}^{\ell}\big(\mathbb{E}_D[p_j^{4\ell}(\mathbf{x})]\big)^{\frac{1}{2\ell}}$. Due to properties (3.30) and (3.31), we finally have the bound $\ell\sqrt{\epsilon_2}\cdot\prod_{j=1}^{\ell}\|p_j\|_{L_{4\ell}}^2 \leq \ell\sqrt{\epsilon_2}(1+\frac{1}{2\ell^2})^{2\ell} \leq 3\ell\sqrt{\epsilon_2}$. Therefore, in total, we have $\|p - h\|_{L_2(D)}^2 \leq 4\ell^2\epsilon_1 + 3\ell^3\epsilon_2 \leq \epsilon$, which implies that $\|p_{\mathrm{up}} - f\|_{\mathcal{L}_2(D)} \leq \epsilon$ and $p_{\mathrm{up}} \geq f$. □

3.10 Lower Bounds

3.10.1 Lower Bound for Realizable TDS Learning of Monotone Functions

We now prove Theorem 17, which we restate here for convenience.

Theorem 23 (Hardness of TDS Learning Monotone Functions)**.** *Let the accuracy parameter ϵ be at most 0.1 and the success probability parameter δ also be at most 0.1. Then, in the realizable setting, any TDS learning algorithm for the class of monotone functions over $\{\pm 1\}^d$ with accuracy parameter ϵ and success probability at least $1-\delta$ requires either $2^{0.04d}$ training samples or $2^{0.04d}$ testing samples for all sufficiently large values of d.*

We will need the following standard fact, see for example Chapter 1 for a proof:

Fact 3. *For any distribution D over any domain, let multisets T_1 and T_2 be sampled as follows:*

1. *Set T_1 is N i.i.d. samples from D.*
2. *First, multiset S is formed by taking M i.i.d. samples from D. Then, multiset T_2 is formed by taking N i.i.d. uniform elements from S.*

Then, the statistical distance between the distributions of T_1 and T_2 is at most $\frac{N^2}{M}$.

Now, we prove Theorem 17.

Proof of Theorem 17. We fix $\delta \leq 0.1$ and also fix $\epsilon \leq 0.1$. Let $\mathcal{A}$ be an algorithm that takes $N \leq 2^{0.04d}$ testing samples and $N \leq 2^{0.04d}$ training samples, and either outputs REJECT, or (ACCEPT, $\widehat{f}$) for a function $\widehat{f} : \{\pm 1\}^d \to \{\pm 1\}$. We argue that for, a sufficiently large d, the algorithm $\mathcal{A}$ will fail to be a TDS-learning algorithm for monotone functions over $\{\pm 1\}^d$.

Let f be some function mapping $\{\pm 1\}^d \to \{\pm 1\}$ and let a multiset S consist of elements in $\{\pm 1\}^d$. We define $\mathcal{T}(f, S)$ to be a random variable supported on $\{\text{Yes}, \text{No}\}$ determined as follows (informally, if $\mathcal{A}$ is a TDS-learner for monotone functions, then $\mathcal{T}(f, S)$ will allow us to distinguish a uniform distribution over S from the uniform distribution over $\{\pm 1\}^d$):

1. Let $S_{\text{train}} \subset \{\pm 1\}^d \times \{\pm 1\}$ consist of N pairs $(\mathbf{x}, f(\mathbf{x}))$, where $\mathbf{x}$ are drawn i.i.d. uniformly from $\{\pm 1\}^d$.
2. Let X_{test} consist of N i.i.d. uniform samples from set S.
3. The algorithm $\mathcal{A}$ is run on $(S_{\text{train}}, X_{\text{test}})$.
4. If $\mathcal{A}$ outputs REJECT, then output $\mathcal{T}(f, S) =$No.
5. If $\mathcal{A}$ outputs (ACCEPT, $\widehat{f}$), then
   (a) Obtain a new set X_2 of 10000 i.i.d. uniform samples from S.
   (b) If, on the majority of points $\mathbf{x}$ in X_2, we have $\widehat{f}(\mathbf{x}) = 1$, then output No.
   (c) Otherwise, output Yes.

For a multiset S consisting of elements in $\{\pm 1\}^d$, let f_S be the monotone function defined as follows:

$$f_S(\mathbf{x}) := \begin{cases} +1 & \text{if there exists } \mathbf{z} \in S : \ \mathbf{x} \succeq \mathbf{z}, \\ -1 & \text{otherwise.} \end{cases}$$

First, we observe that if $\mathcal{A}$ is indeed a (ϵ, δ)-TDS learning algorithm for monotone functions over $\{\pm 1\}^d$, then:

- $\mathcal{T}(-1, \{\pm 1\}^d)$=Yes with probability at least $\frac{2}{3}$ (from here on, by -1 we mean the function that maps every element in $\{\pm 1\}^d$ into -1). This is true because, by the definition of a TDS learner, since S_{train} comes from the uniform distribution over $\{\pm 1\}^d$, with probability at least $1 - 2\delta = 0.8$ the algorithm $\mathcal{A}$ will output (ACCEPT, $\widehat{f}$) for some $\widehat{f}$ satisfying $\mathbb{P}_{\mathbf{x} \sim \{\pm 1\}^d}[\widehat{f}(x) \neq -1] \leq \epsilon = 0.1$. Then, via a standard Hoeffding bound, with probability at least 0.9 on the majority of elements $\mathbf{x}$ in X_2 we have $\widehat{f}(\mathbf{x}) = -1$ and then $\mathcal{T}(-1, \{\pm 1\}^d)$=Yes.
- For any multiset S with elements in $\{\pm 1\}^d$, we have $\mathcal{T}(f_S, S) =$No with probability at least $\frac{2}{3}$. Indeed, from the definition of a TDS learning algorithm, we see that, with probability at least $1 - \delta = 0.9$, the algorithm $\mathcal{A}$ will either output

– Output reject, in which case $\mathcal{T}(f_S, S) =$No.

– Output (ACCEPT, $\widehat{f}$) with $\mathbb{P}_{\mathbf{x}\sim S}[\widehat{f}(\mathbf{x}) \neq f_S(\mathbf{x})] \leq \epsilon = 0.1$. But we know that f_S takes values $+1$ on all elements in S. Therefore, $\mathbb{P}_{\mathbf{x}\sim S}[\widehat{f}(\mathbf{x}) \neq f_S(\mathbf{x})] \leq 0.1$. Then, via a standard Hoeffding bound, with probability at least 0.9 on the majority of elements $\mathbf{x}$ in X_2 we have $\widehat{f}(\mathbf{x}) = +1$ and then $\mathcal{T}(f_S, S)$=No.

In particular, if S is obtained by picking $M = 2^{0.1d}$ i.i.d. elements from $\{\pm 1\}^d$, we have

$$\left| \mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} [\mathcal{T}(f_S, S) = \text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of } \mathcal{T}} [\mathcal{T}(-1, \{\pm 1\}^d) = \text{Yes}] \right| > \frac{1}{3}. \tag{3.32}$$

The rest of the proof argues, via a hybrid argument, that this is impossible. To be specific, we claim that for sufficiently large d the following two inequalities must hold

$$\left| \mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} [\mathcal{T}(-1, S) = \text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of } \mathcal{T}} [\mathcal{T}(-1, \{\pm 1\}^d) = \text{Yes}] \right| \leq \frac{N^2}{M}. \tag{3.33}$$

$$\left| \mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} [\mathcal{T}(f_S, S) = \text{Yes}] - \mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} [\mathcal{T}(-1, S) = \text{Yes}] \right| \leq 2 \left(\frac{3}{4}\right)^d MN. \tag{3.34}$$

We observe that Equation 3.33 follows immediately from Fact 3, because if Equation 3.33 didn't hold, then we would be able to achieve advantage greater than $\frac{M}{N^2}$ when distinguishing N i.i.d. uniform samples from $\{\pm 1\}^d$ from N i.i.d. uniform examples from S.

Now we prove Equation 3.34. Let $S_{\text{train}}^{\mathcal{T}(f_S,S)}$ denote the collection of pairs $\{(\mathbf{x}, f_S(\mathbf{x}))\}$ sampled in Step 1 of $\mathcal{T}(f_S, S)$. Analogously, let $S_{\text{train}}^{\mathcal{T}(-1,S)}$ denote the collection of pairs $(\mathbf{x}, -1)$ in set used in procedure $\mathcal{T}(-1, S)$. In either case, the elements in $S_{\text{train}}^{\mathcal{T}(f_S,S)}$ and $S_{\text{train}}^{\mathcal{T}(-1,S)}$ are i.i.d. uniformly random elements in $\{\pm 1\}^d$. Let $E^{\mathcal{T}(-1,S)}$ be the event, over the choice of S and the choice of $S_{\text{train}}^{\mathcal{T}(-1,S)}$, that for every $(\mathbf{x}, -1) \in S_{\text{train}}^{\mathcal{T}(-1,S)}$ there is no $\mathbf{z}$ in S satisfying $\mathbf{x} \succeq \mathbf{z}$. Analogously, let $E^{\mathcal{T}(f_S,S)}$ be the event, over the choice of S and the choice of $S_{\text{train}}^{\mathcal{T}(f_S,S)}$, that for every $(\mathbf{x}, f_S(\mathbf{x})) \in S_{\text{train}}^{\mathcal{T}(f_S,S)}$ there is no $\mathbf{z}$ in S satisfying $\mathbf{x} \succeq \mathbf{z}$. We observe that

$$\mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} \left[\mathcal{T}(f_S, S) = \text{Yes} \middle| E^{\mathcal{T}(f_S,S)} \right] = \mathop{\mathbb{P}}_{\substack{S\sim \mathrm{Unif}(\{\pm 1\}^d)^{\otimes M} \\ \text{Randomness of } \mathcal{T}}} \left[\mathcal{T}(-1, S) = \text{Yes} \middle| E^{\mathcal{T}(-1,S)} \right] \tag{3.35}$$

which is true because, subject to $E^{\mathcal{T}(f_S,S)}$ or $E^{\mathcal{T}(-1,S)}$, the function f_S takes values of -1 on every element $\mathbf{x}$ in $S_{\text{train}}^{\mathcal{T}(f_S,S)}$ and $S_{\text{train}}^{\mathcal{T}(-1,S)}$ respectively. We also see that the random variables $(S, S_{\text{train}}^{\mathcal{T}(f_S,S)})$ and $(S, S_{\text{train}}^{\mathcal{T}(-1,S)})$ are identically distributed (conditioned on $E^{\mathcal{T}(f_S,S)}$ and $E^{\mathcal{T}(-1,S)}$ respectively).

We also observe that

$$\mathop{\mathbb{P}}_{\substack{S\sim\mathrm{Unif}(\{\pm1\}^d)^{\otimes M}\\ \text{Randomness of }\mathcal{T}}}\left[E^{\mathcal{T}(f_S,S)}\right] = \mathop{\mathbb{P}}_{\substack{S\sim\mathrm{Unif}(\{\pm1\}^d)^{\otimes M}\\ \text{Randomness of }\mathcal{T}}}\left[E^{\mathcal{T}(-1,S)}\right] \leq \left(\frac{3}{4}\right)^d MN, \tag{3.36}$$

where the equality of the two probabilities follows immediately by definition, and the upper bound of $\left(\frac{3}{4}\right)^d MN$ is true for the following reason. Let $\mathbf{z}$ and $\mathbf{x}$ be a pair of i.i.d. uniformly random elements in $\{\pm1\}^d$, then $\mathbb{P}[\mathbf{x} \succeq \mathbf{z}] = \left(\frac{3}{4}\right)^d$ as each bit of $\mathbf{x}$ and $\mathbf{z}$ are independent and for each of the bits we have $\mathbf{x} \geq \mathbf{z}$ with probability exactly $3/4$. Now, taking a union bound over every $(\mathbf{x}, -1) \in S^{\mathcal{T}(-1,S)}_{\text{train}}$ and $\mathbf{z} \in S$, we obtain the bound in Equation 3.36.

Overall, combining Equation 3.33 with Equation 3.34 and substituting $N \leq 2^{0.04d}$ and $M = 2^{0.1d}$ we get

$$\left|\mathop{\mathbb{P}}_{\substack{S\sim\mathrm{Unif}(\{\pm1\}^d)^{\otimes M}\\ \text{Randomness of }\mathcal{T}}}[\mathcal{T}(f_S, S) = \text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of }\mathcal{T}}[\mathcal{T}(-1, \{\pm1\}^d) = \text{Yes}]\right| \leq \frac{N^2}{M} + 2\left(\frac{3}{4}\right)^d MN = 2^{-\Omega(d)},$$

which is in contradiction with Equation 3.32 for a sufficiently large value of d. This proves that $\mathcal{A}$ is not a (ϵ, δ)-TDS learning algorithm for monotone functions. □

3.10.2 Lower Bound for Realizable TDS Learning of Convex Sets

We now prove Theorem 18 which we restate here for convenience.

Theorem 24 (Hardness of TDS Learning Convex Sets)**.** *Let the accuracy parameter ϵ be at most 0.1 and the success probability parameter δ also be at most 0.1. Then, in the realizable setting, any TDS learning algorithm for the class of indicators of convex sets under the standard Gaussian distribution on $\mathbb{R}^d$ requires either $2^{0.04d}$ training samples or $2^{0.04d}$ testing samples for all sufficiently large values of d.*

We will need the following standard facts about Gaussian distributions:

Fact 4 (Concentration of Gaussian norm, see e.g. Lemma 8.1 in [BM97])**.** *For any $\eta > 0$ it is the case that*

$$\mathop{\mathbb{P}}_{\mathbf{x}\in\mathcal{N}(0,I_d)}\left[d - 2\sqrt{d\ln\left(\frac{2}{\eta}\right)} \leq \|\mathbf{x}\|_2^2 \leq d + 2\sqrt{d\ln\left(\frac{2}{\eta}\right)} + 2\ln\left(\frac{2}{\eta}\right)\right] \geq 1 - \eta$$

Fact 5 (Concentration of Gaussian norm. See e.g. Chapter 1.)**.** *For any $r > 0$ it is the case that*

$$\mathop{\mathbb{P}}_{\mathbf{x}^1,\mathbf{x}^2\in\mathcal{N}(0,I_d)}\left[\left\|\mathbf{x}^1-\mathbf{x}^2\right\|_2 \le r\right] \le \left(\frac{64r^2}{d}\right)^{d/2}$$

Recall that we use $\mathcal{B}_a$ to denote the origin-centered closed ball in $\mathbb{R}^d$ of radius a. Using $\mathrm{conv}(\cdot)$ to denote the convex hull of a set of points, will state the following geometric observation in Chapter 1 about convex hulls of a collection of point.

Fact 6. *For any $a > 0$, let $\{\mathbf{x}^i\}_{i=1}^M$ be a collection of points in $\mathcal{B}_b \setminus \mathcal{B}_a$. If for every pair of points $(\mathbf{x}^i, \mathbf{x}^j)$ the $\|\mathbf{x}^i - \mathbf{x}^j\|_2$ is greater than $2\sqrt{b^2-a^2}$, then for every i and j we have*

$$\mathrm{conv}(\mathbf{x}^i, \mathcal{B}_a) \cap \mathrm{conv}(\mathbf{x}^i, \mathcal{B}_a) = \mathcal{B}_a$$

and also

$$\mathrm{conv}(\mathbf{x}^1, \cdots, \mathbf{x}^M, \mathcal{B}_a) = \cup_i \mathrm{conv}(\mathbf{x}^i, \mathcal{B}_a).$$

For the rest of the section we will set

$$a = \sqrt{d - 2\sqrt{d\ln\left(\frac{1}{50}\right)}} \qquad b = \sqrt{d + 2\sqrt{d\ln\left(\frac{1}{50}\right)} + 2\ln\left(\frac{1}{50}\right)}, \tag{3.37}$$

and from Fact 4 we see that the norm a standard Gaussian vector in $\mathbb{R}^d$ falls in interval (a, b) with probability at least 0.99.

Now, we are ready to prove Theorem 18.

Proof of Theorem 17. We fix $\delta \le 0.1$ and also fix $\epsilon \le 0.1$. Let $\mathcal{A}$ be an algorithm that takes $N \le 2^{0.04d}$ testing samples and $N \le 2^{0.04d}$ training samples, and either outputs REJECT, or (ACCEPT, $\widehat{f}$) for a function $\widehat{f} : \mathbb{R}^d \to \{\pm 1\}$. We argue that for, a sufficiently large d, the algorithm $\mathcal{A}$ will fail to be a TDS-learning algorithm for convex sets under the Gaussian distribution on $\mathbb{R}^d$.

For a set S we will define g_S as the indicator of the convex set $\mathrm{conv}(S \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a)$. And in this section we denote the uniform distribution over S as $\mathbb{U}_S$.

Let f be some function mapping $\mathbb{R}^d \to \{\pm 1\}$ and let a set D be a distribution over $\mathbb{R}^d$. We define $\mathcal{H}(f, D)$ to be a random variable supported on $\{\text{Yes}, \text{No}\}$ determined as follows (informally, if $\mathcal{A}$ is a TDS-learner for convex sets, then $\mathcal{H}(f, D)$ will allow us to distinguish D from the Gaussian distribution over $\mathbb{R}^d$):

1. Let $S_{\text{train}} \subset \mathbb{R}^d \times \{\pm 1\}$ consist of N pairs $(\mathbf{x}, f(\mathbf{x}))$, where $\mathbf{x}$ are drawn i.i.d. from $\mathcal{N}(0, I_d)$.
2. Let X_{test} consist of N i.i.d. uniform samples from D.
3. The algorithm $\mathcal{A}$ is run on $(S_{\text{train}}, X_{\text{test}})$.
4. If $\mathcal{A}$ outputs REJECT, then output $\mathcal{H}(f, S) =$No.

5. If $\mathcal{A}$ outputs (ACCEPT, $\widehat{f}$), then
   (a) Obtain a new set X_2 of 10000 i.i.d. samples from D.
   (b) If, on the majority of points $\mathbf{x}$ in X_2, we have $\widehat{f}(\mathbf{x}) = -1$, then output No.
   (c) Otherwise, output Yes.

First, we observe that if $\mathcal{A}$ is indeed a (ϵ, δ)-TDS learning algorithm for convex sets over $\mathbb{R}^d$ under $\mathcal{N}(0, I_d)$, then:

- $\mathcal{H}(g_\emptyset, \mathcal{N}(0, I_d))$=Yes with probability at least $\frac{2}{3}$ (from here on, by -1 we mean the function that maps every element in $\{\pm 1\}^d$ into -1). This is true because, by the definition of a TDS learner, since S_{train} comes from the uniform distribution over $\mathcal{N}(0, I_d)$, with probability at least $1 - 2\delta = 0.8$ the algorithm $\mathcal{A}$ will output (ACCEPT, $\widehat{f}$) for some $\widehat{f}$ satisfying $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}_d}[\widehat{f}(\mathbf{x}) \neq g_\emptyset(\mathbf{x})] \leq \epsilon = 0.1$. Since a was chosen is such manner that $\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[\mathbf{x} \in \mathcal{B}_a] < 0.01$, and $g_\emptyset$ is the indicator function of $\mathcal{B}_a$, we have $\mathbb{P}_{\mathbf{x}\in\mathcal{N}(0,I_d)}[g_\emptyset(\mathbf{x}) \neq -1] < 0.01$. Via a union bound, we see that $\mathbb{P}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}[\widehat{f}(\mathbf{x}) \neq -1] \leq 0.11$. Then, via a standard Hoeffding bound, with probability at least 0.9 on the majority of elements $\mathbf{x}$ in X_2 we have $\widehat{f}(\mathbf{x}) = -1$ and then $\mathcal{H}(g_\emptyset, \mathcal{N}(0, I_d))$=Yes.
- For any set S with elements in $\mathbb{R}^d$, we have $\mathcal{H}(g_S, \mathbb{U}_S)$ =No with probability at least $\frac{2}{3}$. Indeed, from the definition of a TDS learning algorithm, we see that, with probability at least $1 - \delta = 0.9$, the algorithm $\mathcal{A}$ will either
  - Output reject, in which case $\mathcal{H}(g_S, \mathbb{U}_S)$ =No.
  - Output (ACCEPT, $\widehat{f}$) with $\mathbb{P}_{\mathbf{x}\sim\mathbb{U}_S}[\widehat{f}(\mathbf{x}) \neq g_S(\mathbf{x})] \leq \epsilon = 0.1$. But we know that g_S takes values $+1$ on all elements in S. Therefore, $\mathbb{P}_{\mathbf{x}\sim\mathbb{U}_S}[\widehat{f}(\mathbf{x}) \neq f_S(\mathbf{x})] \leq 0.1$. Then, via a standard Hoeffding bound, with probability at least 0.9 on the majority of elements $\mathbf{x}$ in X_2 we have $\widehat{f}(\mathbf{x}) = +1$ and then $\mathcal{H}(g_S, \mathbb{U}_S)$=No.

In particular, if S is obtained by picking $M = 2^{0.1d}$ i.i.d. elements from $\mathcal{N}(0, I_d)$, we have

$$\left| \mathop{\mathbb{P}}_{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M} \\ \text{Randomness of } \mathcal{H}}} [\mathcal{H}(g_S, \mathbb{U}_S) = \text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of } \mathcal{H}} [\mathcal{H}(g_\emptyset, \mathcal{N}(0, I_d)) = \text{Yes}] \right| > \frac{1}{3}. \tag{3.38}$$

The rest of the proof argues, via a hybrid argument, that this is impossible. To be specific, we claim that for sufficiently large d the following two inequalities must hold

$$\left| \mathop{\mathbb{P}}_{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M} \\ \text{Randomness of } \mathcal{H}}} [\mathcal{H}(g_\emptyset, \mathbb{U}_S) = \text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of } \mathcal{H}} [\mathcal{H}(g_\emptyset, \mathcal{N}(0, I_d)) = \text{Yes}] \right| \leq \frac{N^2}{M}. \tag{3.39}$$

$$\left| \underset{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of } \mathcal{H}}}{\mathbb{P}}[\mathcal{H}(g_S, \mathbb{U}_S) = \text{Yes}] - \underset{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of } \mathcal{H}}}{\mathbb{P}}[\mathcal{H}(g_\emptyset, \mathbb{U}_S) = \text{Yes}] \right| \leq \left(\frac{64(b^2-a^2)}{d}\right)^{d/2} (M+N)^2. \tag{3.40}$$

We observe that Equation 3.39 follows immediately from Fact 3, because if Equation 3.39 didn't hold, then we would be able to achieve advantage greater than $\frac{M}{N^2}$ when distinguishing N i.i.d. uniform samples from $\mathcal{N}(0, I_d)$ and N i.i.d. uniform examples from S.

Now we prove Equation 3.40. Let $S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ denote the collection of pairs $\{(\mathbf{x}, g_S(\mathbf{x}))\}$ sampled in Step 1 of $\mathcal{H}(g_S, \mathbb{U}_S)$. Analogously, let $S_{\text{train}}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$ denote the collection of pairs $(\mathbf{x}, -1)$ in set used in procedure $\mathcal{H}(g_\emptyset, \mathbb{U}_S)$. In either case, the elements in $S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ and $S_{\text{train}}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$ are i.i.d. elements from $\mathcal{N}(0, I_d)$. Let $\mathcal{E}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ be the event, over the choice of S and the choice of $S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}$, that for each pair of points $\mathbf{x}^1$ and $\mathbf{x}^2$ in $S \cup \{\mathbf{x} : \ (\mathbf{x}, g_S(x)) \in S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}\}$ we have $\|\mathbf{x}^1 - \mathbf{x}^2\|_2 > 2\sqrt{b^2-a^2}$. Analogously, let $\mathcal{E}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ be the event, over the choice of S and the choice of $S_{\text{train}}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$, that for each pair of points $\mathbf{x}^1$ and $\mathbf{x}^2$ in $S \cup \{\mathbf{x} : \ (\mathbf{x}, g_\emptyset(x)) \in S_{\text{train}}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}\}$ we have $\|\mathbf{x}^1 - \mathbf{x}^2\|_2 > 2\sqrt{b^2-a^2}$.

We first observe that subject to $\mathcal{E}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$ it is the case that for every $\{(\mathbf{x}, g_S(\mathbf{x}))\}$ in $S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ it is the case that $g_S = g_\emptyset(x)$. For $\mathbf{x} \in \mathcal{B}_a \cup (\mathbb{R} \setminus \mathcal{B}_b)$ this is immediate because g_S as the indicator of the convex set $\text{conv}(S \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a)$. It remains to show this only for points $(\mathbf{x}, g_S(\mathbf{x})) \in S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ that also satisfy $\mathbf{x} \in \mathcal{B}_b \setminus \mathcal{B}_a$. Since $\mathbf{x}$ is outside $\mathcal{B}_a$, we have $g_\emptyset(\mathbf{x}) = -1$ and therefore we would like to show that $g_S(\mathbf{x})$ also equals to -1. This is true because from Fact 6 it is the case that if $\mathcal{E}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$ takes place, then for every such $\mathbf{x}$ we have

$$\text{conv}(\mathbf{x}, \mathcal{B}_a) \cap \text{conv}(S \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a) = \text{conv}(\mathbf{x}, \mathcal{B}_a) \cap \left(\bigcup_{\mathbf{z} \in S \cap (\mathcal{B}_b \setminus \mathcal{B}_a)} \text{conv}(\mathbf{z} \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a) \right) =$$
$$\bigcup_{\mathbf{z} \in S \cap (\mathcal{B}_b \setminus \mathcal{B}_a)} (\text{conv}(\mathbf{x}, \mathcal{B}_a) \cap (\text{conv}(\mathbf{z} \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a))) = \mathcal{B}_a,$$

which in particular implies that $\mathbf{x}$ is not in the convex hull $\text{conv}(S \cap (\mathcal{B}_b \setminus \mathcal{B}_a), \mathcal{B}_a)$ and $g_S(\mathbf{x}) = -1$, concluding the proof of our observation.

We therefore conclude that distributions of $(S, S_{\text{train}}^{\mathcal{H}(g_S,\mathbb{U}_S)})$ and $(S, S_{\text{train}}^{\mathcal{H}(\mathcal{H}(g_\emptyset,\mathbb{U}_S)})$ are identically distributed conditioned on $\mathcal{E}^{\mathcal{H}(g_S,\mathbb{U}_S)}$ and $\mathcal{E}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}$ respectively, which implies that

$$\underset{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of } \mathcal{H}}}{\mathbb{P}} \left[\mathcal{H}(g_S, \mathbb{U}_S) = \text{Yes} \middle| \mathcal{E}^{\mathcal{H}(g_S,\mathbb{U}_S)} \right] = \underset{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of } \mathcal{H}}}{\mathbb{P}} \left[\mathcal{H}(g_\emptyset, \mathbb{U}_S) = \text{Yes} \middle| \mathcal{E}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)} \right], \tag{3.41}$$

We also observe that

$$\mathop{\mathbb{P}}_{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of }\mathcal{H}}}\left[\mathcal{E}^{\mathcal{H}(g_S,\mathbb{U}_S)}\right] = \mathop{\mathbb{P}}_{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of }\mathcal{H}}}\left[\mathcal{E}^{\mathcal{H}(g_\emptyset,\mathbb{U}_S)}\right] \leq \left(\frac{64(b^2-a^2)}{d}\right)^{d/2}(M+N)^2, \quad (3.42)$$

where the equality of the two probabilities follows immediately by definition, and the upper bound of $\left(\frac{64(b^2-a^2)}{d}\right)^{d/2}(M+N)^2$ is true by applying Fact 5 to each relevant pair of points. Therefore, we obtain the bound in Equation 3.42.

Overall, combining Equation 3.39 with Equation 3.40 and substituting $N \leq 2^{0.04d}$, $M = 2^{0.1d}$ as well as $a = \sqrt{d - 2\sqrt{d\ln\left(\frac{1}{50}\right)}}$ and $b = \sqrt{d + 2\sqrt{d\ln\left(\frac{1}{50}\right)} + 2\ln\left(\frac{1}{50}\right)}$, we obtain

$$\left|\mathop{\mathbb{P}}_{\substack{S\sim\mathcal{N}(0,I_d)^{\otimes M}\\ \text{Randomness of }\mathcal{H}}}[\mathcal{H}(f_S,S)=\text{Yes}] - \mathop{\mathbb{P}}_{\text{Randomness of }\mathcal{H}}[\mathcal{H}(g_\emptyset,\mathcal{N}(0,I_d))=\text{Yes}]\right| \leq$$
$$\frac{N^2}{M} + \left(\frac{64(b^2-a^2)}{d}\right)^{d/2}(M+N)^2 = 2^{-0.02d} + \left(O\left(\frac{1}{\sqrt{d}}\right)\right)^{d/2} = 2^{-\Omega(d)},$$

which is in contradiction with Equation 3.38 for a sufficiently large value of d. This proves that $\mathcal{A}$ is not a (ϵ,δ)-TDS learning algorithm for convex sets. □

3.10.3 Lower Bound for the Agnostic Error Guarantee

We now focus on the agnostic setting and provide an information theoretic lower bound on the error upon acceptance. Our lower bound is simple and demonstrates that a linear dependence on the error factor λ (see Equation (3.2)) is unavoidable for TDS learning.

Theorem 25 (Lower Bound for the Error in the Agnostic Setting)**.** *Let $\mathcal{X}$ be any domain, D a distribution over $\mathcal{X}$ and $\mathcal{C}$ a class of concepts that map $\mathcal{X}$ to $\{\pm 1\}$ that is closed under complement, i.e., if $f \in \mathcal{C}$ then $-f \in \mathcal{C}$. Then, for any $\eta \in (0,1/2)$, any $\epsilon \in (0,\eta/2)$ and $\delta \in (0,1/3)$, no TDS learning algorithm for $\mathcal{C}$ w.r.t. D with finite sample complexity and failure probability δ, can have an error guarantee better than $\lambda(1-2\eta)+\epsilon = \Omega(\lambda)+\epsilon$.*

Proof. Let $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ denote the training distribution and $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$ the test distribution, which are both over $\mathcal{X}\times\{\pm 1\}$. Suppose that for $\eta \in (0,1/2)$ and $\epsilon \in (0,\eta/2)$ there exists an algorithm $\mathcal{A}$, that, upon acceptance and with probability at least $1-\delta$, outputs $\widehat{f} \in \mathcal{C}$ with $\text{err}(\widehat{f};\mathcal{D}^{\text{test}}_{\mathcal{XY}}) \leq \lambda(1-2\eta)+\epsilon$ ($\lambda = \lambda(\mathcal{C};\mathcal{D}^{\text{train}}_{\mathcal{XY}},\mathcal{D}^{\text{test}}_{\mathcal{XY}})$, see Equation (3.2)). Let $C > 0$ be a sufficiently large universal constant.

We consider the following algorithm $\mathcal{T}$. Algorithm $\mathcal{T}$ uses an oracle to $\mathcal{A}$ and accepts or rejects according to the following criteria.

- If $\mathcal{A}$ rejects, then $\mathcal{T}$ rejects.

- If $\mathcal{A}$ accepts and outputs $\widehat{f} \in \mathcal{C}$, then $\mathcal{T}$ draws $\frac{C}{\eta^2}\log(1/\delta)$ examples $S_{\mathcal{T}}$ from $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ and rejects if $\mathbb{P}_{(\mathbf{x},y)\in S_{\mathcal{T}}}[\widehat{f}(\mathbf{x}) \neq y] > 3\eta/4$. Otherwise, $\mathcal{T}$ accepts.

Fix some $f \in \mathcal{C}$ and let $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ be the distribution over $\mathcal{X} \times \{\pm 1\}$ whose marginal on $\mathcal{X}$ is D and the labels are generated as $y(\mathbf{x}) = f(\mathbf{x})$. Consider the following two cases about $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$.

Case 1. First, suppose that $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ has D as marginal on $\mathcal{X}$ and $y(\mathbf{x}) = f(\mathbf{x})$. Then, $\mathcal{A}$ accepts with probability at least $1-\delta$, due to completeness. We have $\lambda = 0$ (attained by f) and, hence, upon acceptance, $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y] = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y] \leq \epsilon \leq \eta/2$ with probability at least $1-\delta$. By a Hoeffding bound, we then have that $\mathcal{T}$ must accept with probability at least $1-\delta$. Overall, $\mathcal{T}$ accepts with probability at least $1-3\delta > 1/2$.

Case 2. Second, suppose that $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ has D as marginal on $\mathcal{X}$ and $y(\mathbf{x}) = -f(\mathbf{x})$. Then, we have that $\lambda = 1$ (because for any point $\mathbf{x} \in \mathcal{X}$, any classifier will either classify $\mathbf{x}$ incorrectly under $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}$ or under $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$). By assumption, we have $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y] \leq \lambda(1-2\eta)+\epsilon \leq 1-2\eta+\epsilon$ with probability at least $1-2\delta$ (by completeness and soundness). Since the test labels are the negation of the train labels, we have $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y] = 1 - \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y]$, and $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}}[\widehat{f}(\mathbf{x}) \neq y] \geq 2\eta - \epsilon \geq \eta$ (since $\epsilon \leq \eta/2$). By a Hoeffding bound, $\mathcal{T}$ will reject with probability at least $1-3\delta > 1/2$.

We have reached a contradiction, because in both cases, the input of $\mathcal{T}$ does not depend on the test labels, and everything else remains the same in both cases. Therefore, $\mathcal{T}$ should have the same behavior in both cases and we conclude that the algorithm $\mathcal{A}$ cannot exist as defined. □

Remark 4. *While the above lower bound demonstrates that the error of a TDS learning algorithm can be necessarily high in certain settings, we emphasize that the construction corresponds to a contrived case where the training distribution does not provide enough information about the test distribution and, therefore, any meaningful notion of learning should be hopeless (see also [BDU12]).*

3.11 Sample Complexity of TDS Learning

In the previous sections, we explored a number of computational aspects of TDS learning, deriving dimension efficient algorithms for several instantiations of our setting. In this section, we focus on the statistical aspects of TDS learning. There are several prior works in the literature of domain adaptation that study the statistical landscape of the problem of learning under shifting distributions (see, e.g., [BDBCP06, BCK$^+$07, MMR09, BDBC$^+$10, DLLP10]). All of the previous generalization upper bounds on this problem involve some discrepancy term, which quantifies the amount of distribution shift, as well as some additional terms that are typically considered small for reasonable settings. For a concept class $\mathcal{C} : \mathcal{X} \to \{\pm 1\}$, considering that the error term λ (see Eq. (3.2)) is small is a standard assumption in domain adaptation (see, e.g., [BDBCP06, BCK$^+$07]). Furthermore, one standard measure of discrepancy is defined as follows.

Definition 14 (Discrepancy Distance, [BCK+07]). *Let $\mathcal{X} \subset \mathbb{R}^d$ and let $\mathcal{C}$ be a concept class mapping $\mathcal{X}$ to $\{\pm 1\}$. For distributions D, D' over $\mathcal{X}$, we define the discrepancy distance* $\mathrm{disc}_{\mathcal{C}}(D, D')$ *as follows.*

$$\mathrm{disc}_{\mathcal{C}}(D, D') = \sup_{f, f' \in \mathcal{C}} \left| \mathbb{P}_{D}[f(\mathbf{x}) \neq f'(\mathbf{x})] - \mathbb{P}_{D'}[f(\mathbf{x}) \neq f'(\mathbf{x})] \right|$$

In particular, [BDBCP06, BCK+07] observe that for any $f \in \mathcal{C}$ and distributions $\mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$ over $\mathcal{X} \times \{\pm 1\}$ the following is true.

$$\mathrm{err}(f; \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \leq \mathrm{err}(f; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}) + \mathrm{disc}_{\mathcal{C}}(\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{X}}) + \lambda(\mathcal{C}; \mathcal{D}^{\mathrm{train}}_{\mathcal{XY}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \tag{3.43}$$

The bound of Eq. (3.43) can be translated to a generalization bound for domain adaptation, through the use Rademacher complexity, whose definition is provided below.

Definition 15 (Rademacher Complexity). *Let $\mathcal{X} \subseteq \mathbb{R}^d$, let D be a distribution over $\mathcal{X}$ and let $\mathcal{C}$ be a concept class mapping $\mathcal{X}$ to $\{\pm 1\}$. For a set of m samples $X = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)})$ drawn independently from D, we define the empirical Rademacher complexity of $\mathcal{C}$ w.r.t. X as follows*

$$\widehat{\mathfrak{R}}_X(\mathcal{C}) = \frac{2}{m} \mathbb{E} \sup_{f \in \mathcal{C}} \sum_{j=1}^{m} \sigma_j f(\mathbf{x}^{(j)}) \,, \text{ where the expectation is over } \sigma \sim \mathrm{Unif}(\{\pm 1\}^d)$$

Moreover, we define the Rademacher complexity of $\mathcal{C}$ at m w.r.t. D as $\mathfrak{R}_m(\mathcal{C}; D) = \mathbb{E}[\widehat{\mathfrak{R}}_X(\mathcal{C})]$, where the expectation is over $X \sim D^{\otimes m}$.

Corollaries 6, 7 in [MMR09], demonstrate that the discrepancy between two distributions is upper bounded as follows.

Proposition 42 (Bounding the Discrepancy, Corollary 7 in [MMR09]). *Consider $\mathcal{X} \subseteq \mathbb{R}^d$, a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}^d\}$ and distributions D, D' over $\mathcal{X}$. Then for any $\delta > 0$, $m, m' \in \mathbb{N}$, if X, X' are independent examples from D, D', respectively, of sizes m, m', the following is true.*

$$\mathrm{disc}_{\mathcal{C}}(D, D') \leq \mathrm{disc}_{\mathcal{C}}(X, X') + 4\widehat{\mathfrak{R}}_X(\mathcal{C}) + 4\widehat{\mathfrak{R}}_{X'}(\mathcal{C}) + 3\left(\log(4/\delta)\right)^{1/2} \sqrt{\frac{1}{m} + \frac{1}{m'}}$$

Combining inequality (3.43) with Proposition 42 and standard generalization bounds for classification, yields a data-dependent generalization bound for domain adaptation whose only unknown parameter is λ. In our setting this readily implies the following sample complexity upper bound in terms of the Rademacher complexity of the concept class $\mathcal{C}$.

Corollary 5 (Sample Complexity upper bound for TDS learning). *Let $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}\}$ be a hypothesis class and D a distribution over $\mathcal{X}$ such that $\mathfrak{R}_m(\mathcal{C}; D) \leq \epsilon/10$. The algorithm that runs the Empirical Risk Minimizer on training data and accepts only when both the empirical discrepancy distance between the training and test unlabelled examples, i.e.* $\mathrm{disc}_{\mathcal{C}}(X_{\mathrm{train}}, X_{\mathrm{test}})$*, and the Rademacher complexity with respect to the test examples, i.e. $\widehat{\mathfrak{R}}_{X_{\mathrm{test}}}(\mathcal{C})$, are $O(\epsilon)$, is an (ϵ, δ)-TDS learning algorithm for $\mathcal{C}$ up to error $2\lambda + \epsilon$ with sample complexity $O(m + \frac{1}{\epsilon^2}\log(1/\delta))$. Moreover, if there is a concept in $\mathcal{C}$ with zero training error, the same is true up to error $\lambda + \epsilon$.*

We emphasize that, while Corollary 5 readily follows from prior results in the literature of domain adaptation, it highlights an important distinction between domain adaptation and TDS learning: A TDS learning algorithm, upon acceptance, achieves error that does not scale with the discrepancy between the training and test marginal distributions, but only a term that depends on the quantity λ, which, as we show in Theorem 25, is unavoidable.

3.12 PQ Learning and Distribution-Free TDS Learning

In recent years, there has been a vast amount of work on the problem of learning under shifting distributions. One of the most relevant models to TDS learning is PQ learning (see [GKKM20, KK21]), which was defined by [GKKM20]. In this section, we establish a connection between PQ learning and TDS learning and, in particular, we show that TDS learning can be reduced to PQ learning, thereby inheriting all of the existing results in the latter framework. Unfortunately, to the best of our knowledge, most of the positive results on the PQ learning framework make strong assumptions regarding oracle access to solvers of learning primitives that are typically hard to solve. Nonetheless, PQ learning is an important theoretical framework for learning under arbitrary covariate shifts and it is an interesting open question whether our methods can be extended to provide positive results for the not-easier problem of PQ learning.

In the PQ learning framework, a learner outputs a pair $(h, \mathbf{X})$, where $h : \mathcal{X} \to \{\pm 1\}$ is a classifier and $\mathbf{X} \subseteq \mathcal{X}$ is a subset of the feature space where one can be confident on the predictions of h. In particular, the PQ learning model is defined as follows.

Definition 16 (PQ Learning, [GKKM20, KK21])**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a set and $\mathcal{C} \subseteq \mathcal{X} \to \{\pm 1\}$ a concept class. For $\epsilon, \delta \in (0,1)$ we say that algorithm $\mathcal{A}$ PQ learns $\mathcal{C}$ up to error ϵ and probability of failure δ if for any distributions $\mathcal{D}^{\text{train}}_{\mathcal{XY}}, \mathcal{D}^{\text{test}}_{\mathcal{XY}}$ over $\mathcal{X} \times \{\pm 1\}$ such that there is some $f^* \in \mathcal{C}$ so that $y = f^*(\mathbf{x})$ for any $(\mathbf{x}, y)$ drawn from either $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ or $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$, algorithm $\mathcal{A}$, upon receiving a large enough number of labelled samples from $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ and a large enough number of unlabelled samples from $\mathcal{D}^{\text{test}}_{\mathcal{X}}$, outputs a pair $(h, \mathbf{X})$ such that $h : \mathcal{X} \to \{\pm 1\}$, $\mathbf{X} \subseteq \mathcal{X}$ and with probability at least $1 - \delta$ the following is true.*

$$\mathop{\mathbb{P}}_{\mathbf{x} \sim \mathcal{D}^{\text{train}}_{\mathcal{X}}}[\mathbf{x} \notin \mathbf{X}] \leq \epsilon \text{ and } \mathop{\mathbb{P}}_{(\mathbf{x},y) \sim \mathcal{D}^{\text{test}}_{\mathcal{XY}}}[h(\mathbf{x}) \neq y \text{ and } \mathbf{x} \in \mathbf{X}] \leq \epsilon$$

We note that the above definition of PQ learning is distribution-free, i.e., the guarantees hold for any distribution and not with respect to a specific target distribution. In Definition 12 for TDS learning, the completeness criterion is stated with respect to a particular target distribution that is the same as the training distribution. However, in order to demonstrate a connection between PQ learning and TDS learning, we now define Distribution-Free TDS learning.

Definition 17 (Distribution-free TDS Learning))**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ and consider a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \{\pm 1\}\}$. For $\epsilon, \delta \in (0,1)$, we say that an algorithm $\mathcal{A}$ testably learns $\mathcal{C}$ under distribution shifts up to error ϵ and probability of failure δ if the following is true. For any distributions $\mathcal{D}^{\text{train}}_{\mathcal{XY}}, \mathcal{D}^{\text{test}}_{\mathcal{XY}}$ over $\mathcal{X} \times \{\pm 1\}$ such that there is some $f^* \in \mathcal{C}$ such that $y = f^*(\mathbf{x})$ for*

any $(\mathbf{x}, y)$ *drawn from either* $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ *or* $\mathcal{D}^{\text{test}}_{\mathcal{XY}}$*, algorithm* $\mathcal{A}$*, upon receiving a large enough set of labelled samples* S_{train} *from the training distribution* $\mathcal{D}^{\text{train}}_{\mathcal{XY}}$ *and a large enough set of unlabelled samples* X_{test} *from the test distribution* $\mathcal{D}^{\text{test}}_{\mathcal{X}}$*, either rejects* $(S_{\text{train}}, X_{\text{test}})$ *or accepts and outputs a hypothesis* $h : \mathcal{X} \to \{\pm 1\}$ *with the following guarantees.*

(*a*) *(Soundness.) With probability at least* $1 - \delta$ *over the samples* $S_{\text{train}}, X_{\text{test}}$ *we have:*
If A *accepts, then the output* h *satisfies* $\text{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) \leq \epsilon$*.*

(*b*) *(Completeness.) Whenever* $\mathcal{D}^{\text{test}}_{\mathcal{X}} = \mathcal{D}^{\text{train}}_{\mathcal{X}}$*,* A *accepts with probability at least* $1 - \delta$ *over the samples* $S_{\text{train}}, X_{\text{test}}$*.*

We are now ready to prove that distribution-free TDS learning reduces to PQ learning.

Proposition 43 (TDS learning via PQ learning)**.** *Algorithm 6 reduces TDS to PQ learning. In particular, for* $\epsilon, \delta \in (0, 1)$*, PQ learning algorithm* $\mathcal{A}$ *and a concept class* $\mathcal{C}$*, Algorithm 6, upon receiving* $m_P + \frac{C}{\epsilon^2}\log(1/\delta)$ *labelled examples* S_{train} *from the training distribution and* $m_Q + \frac{C}{\epsilon^2}\log(1/\delta)$ *unlabelled examples* X_{test} *from the test distribution where* m_P, m_Q *are such that* $\mathcal{A}$ *is an* $(\epsilon/4, \delta)$*-PQ learning algorithm for* $\mathcal{C}$ *given* m_P *training and* m_Q *test examples,* (ϵ, δ)*-TDS learns* $\mathcal{C}$*.*

Proof. Let $C > 0$ be a sufficiently large universal constant. For **soundness**, we observe that upon acceptance, we have $\mathbb{P}_{\mathbf{x}\sim X_2}[\mathbf{x} \notin \mathbf{X}]$ and by a Hoeffding bound, since $m_2 \geq \frac{C}{\epsilon_2}\log(1/\delta)$, we have $\mathbb{P}_{\mathbf{x}\sim \mathcal{D}^{\text{test}}_{\mathcal{X}}}[\mathbf{x} \notin \mathbf{X}] \leq 2\epsilon/3$. By using the fact that $\text{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) \leq \mathbb{P}_{\mathbf{x}\sim \mathcal{D}^{\text{test}}_{\mathcal{X}}}[\mathbf{x} \in \mathbf{X}] + \mathbb{P}_{\mathbf{x}\sim \mathcal{D}^{\text{test}}_{\mathcal{X}}}[\mathbf{x} \in \mathbf{X}]$ and the guarantee of the PQ learner we obtain $\text{err}(h; \mathcal{D}^{\text{test}}_{\mathcal{XY}}) \leq \epsilon$, with probability at least $1 - \delta$. For **completeness**, we use the definition of PQ learning and a Hoeffding bound to show that with probability at least $1 - \delta$, Algorithm 6 accepts whenever $\mathcal{D}^{\text{test}}_{\mathcal{X}} = \mathcal{D}^{\text{train}}_{\mathcal{X}}$. □

Algorithm 6: TDS learning through PQ learning

Input: Sets S_{train}, X_{test}, parameters $\epsilon, \delta \in (0, 1)$, $(\epsilon' = \frac{\epsilon}{4}, \delta)$-PQ learner $\mathcal{A}$
Set $m_1 = m_Q$, $m_2 = \frac{C}{\epsilon^2}\log(1/\delta)$ and split X_{test} in X_1, X_2 with sizes m_1, m_2.
Run algorithm $\mathcal{A}$ on (S_{train}, X_1) and receive output $(h, \mathbf{X})$.
Reject if $\mathbb{P}_{\mathbf{x}\sim X_2}[\mathbf{x} \notin \mathbf{X}] > \epsilon/3$.
Otherwise, output h and terminate.

The simple reduction we provided in Proposition 43 implies that all of the positive results on PQ learning transfer to TDS learning. Moreover, note that the reduction does not alter the training and test distributions between the corresponding TDS and PQ algorithms and, therefore, would hold even in the distribution specific setting. This is not true, however, about the following corollary which is based on a reduction from PQ learning to reliable agnostic learning, which does not preserve the marginal distributions.

Corollary 6 (Combination of Theorem 5 in [KK21] and Proposition 43)**.** *If a concept class* $\mathcal{C}$ *is distribution -free reliably learnable, then it is TDS learnable in the distribution-free setting.*

We remark that, in fact, (distribution-free) PQ learning is equivalent to (distribution-free) reliable learning (see Theorems 5, 6 in [KK21]). For a definition of reliable learning we refer the reader to [KKM12]. It is known that reliable learning is no harder than agnostic learning and no easier than PAC learning.

3.13 Amplifying success probability

We will now demonstrate that it is possible to amplify the probability of success of a TDS learner through repetition. Note that this is not immediate for TDS learning as it is, for example, in agnostic learning, where one may repeat an agnostic learning algorithm and choose the hypothesis with the smallest error estimate among the outputs of the independent runs. The main obstacle is that test labels are not available. Nonetheless, we obtain the following theorem regarding amplifying the probability of success.

Proposition 44 (Amplifying Success Probability)**.** *Let $\mathcal{C}$ be a hypothesis class, D a distribution and suppose $\mathcal{A}$ is a TDS learner for $\mathcal{C}$ with respect to D with error guarantee $\psi(\lambda) + \epsilon$ and failure probability at most 0.1. Then, there is a TDS learner $\mathcal{A}'$ for $\mathcal{C}$ with respect to D with error guarantee $4\psi(\lambda) + 4\epsilon$ and failure probability at most δ. In particular, $\mathcal{A}'$ repeats $\mathcal{A}$ for $T = O(\log(\frac{1}{\epsilon\delta}))$ times and rejects if most of the repetitions reject. If most repetitions accept, $\mathcal{A}'$ outputs the hypothesis $h = \mathrm{maj}(h_1, \ldots, h_{T/2})$ (h outputs the majority vote of h_i), where $h_1, \ldots, h_{T/2}$ are the outputs of the first $T/2$ repetitions of $\mathcal{A}$ that accepted.*

Proof. We split the proof into two parts, one for soundness and one for completeness.

Soundness. For soundness, suppose that $\mathcal{A}'$ accepts. We denote with $\widehat{\mathbb{P}}$ (resp. $\widehat{\mathbb{E}}$) the probabilities (resp. expectations) over the randomness of $h_1, \ldots, h_{T/2}$ (which originates to the randomness of the samples given to $\mathcal{A}$) and with $\mathbb{P}$ (resp. $\mathbb{E}$) the probabilities (resp. expectations) over the randomness of a pair $(\mathbf{x}, y)$ drawn from $\mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}$. In what follows, let $\eta = \psi(\lambda) + \epsilon$. We have that for any $i = 1, 2, \ldots, T/2$, $\widehat{\mathbb{P}}[\mathrm{err}(h_i, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \leq \eta] \geq 0.9$, by the guarantees of $\mathcal{A}'$. We will show that $\widehat{\mathbb{P}}[\mathrm{err}(h, \mathcal{D}^{\mathrm{test}}_{\mathcal{XY}}) \leq 4\eta] \geq 1 - \delta$ for a sufficiently large $T = O(\log(\frac{1}{\epsilon\delta}))$.

We define $\mathcal{G}_i$ to be the event (over the randomness of h_i) that h_i is 'good', i.e., that $\mathbb{P}[h_i(\mathbf{x}) \neq y] \leq \eta$. We define $\mathbf{Z}$ to be the 'bad' region of $(\mathbf{x}, y)$, i.e., $\mathbf{Z} = \{(\mathbf{x}, y) \in \mathcal{X} \times \{\pm 1\} : \widehat{\mathbb{P}}[h_1(\mathbf{x}) \neq y | \mathcal{G}_1] > 1/3\}$. Note that $\mathbf{Z}$ would be the same even if we substituted $(h_1, \mathcal{G}_1)$ above with an arbitrary $(h_i, \mathcal{G}_i)$.

First, we observe that $\mathbb{P}[h(\mathbf{x}) \neq y] \leq \mathbb{P}[(\mathbf{x}, y) \in \mathbf{Z}] + \mathbb{P}[h(\mathbf{x}) \neq y | (\mathbf{x}, y) \notin \mathbf{Z}]$.

We now observe that $\mathbb{P}[(\mathbf{x}, y) \in \mathbf{Z}] = \mathbb{P}[\widehat{\mathbb{P}}[h_1 \neq y | \mathcal{G}_1] > 1/3] \leq 3\mathbb{E}\widehat{\mathbb{P}}[h_1(\mathbf{x}) \neq y | \mathcal{G}_1]$ by Markov's inequality. Now, we may swap the expectations to obtain $\mathbb{P}[(\mathbf{x}, y) \in \mathbf{Z}] \leq 3\widehat{\mathbb{E}}[\mathbb{P}[h_1(\mathbf{x}) \neq y] | \mathcal{G}_1] \leq 3\eta$.

So far, we have shown $\mathbb{P}[h(\mathbf{x}) \neq y] \leq 3\eta + \mathbb{P}[h(\mathbf{x}) \neq y | (\mathbf{x}, y) \notin \mathbf{Z}]$. We will bound the probability over $h_1, \ldots, h_{T/2}$ that $\mathbb{P}[h(\mathbf{x}) \neq y | (\mathbf{x}, y) \notin \mathbf{Z}] > \eta$. In particular, we have the following

due to Markov's inequality $\widehat{\mathbb{P}}[\mathbb{P}[h(\mathbf{x}) \neq y|(\mathbf{x},y) \notin \mathbf{Z}] > \eta] \leq \frac{1}{\eta}\widehat{\mathbb{E}}[\mathbb{P}[h(\mathbf{x}) \neq y|(\mathbf{x},y) \notin \mathbf{Z}]]$. Once more, we may swap the expectations to obtain $\widehat{\mathbb{P}}[\mathbb{P}[h(\mathbf{x}) \neq y|(\mathbf{x},y) \notin \mathbf{Z}] > \eta] \leq \frac{1}{\eta}\mathbb{E}[\widehat{\mathbb{P}}[h(\mathbf{x}) \neq y]|(\mathbf{x},y) \notin \mathbf{Z}]$.

Moreover, if we fix $(\mathbf{x},y) \notin \mathbf{Z}$, then $\widehat{\mathbb{P}}[h_i(\mathbf{x}) = y] \geq \widehat{\mathbb{P}}[h_i(\mathbf{x}) = y \text{ and } \mathcal{G}_i] \geq \frac{2}{3} \cdot \frac{9}{10} \geq 3/5$. Because $\widehat{\mathbb{P}}[\mathcal{G}_i] \geq 0.9$ and $\widehat{\mathbb{P}}[h_i(\mathbf{x}) = y|\mathcal{G}_i] \geq 2/3$ whenever $(\mathbf{x},y) \notin \mathbf{Z}$, by the definition of $\mathbf{Z}$. Therefore, since $h_1, \ldots, h_{T/2}$ are independent, we have that $\widehat{\mathbb{P}}[h(\mathbf{x}) \neq y] \leq \exp(-T/C)$ for some sufficiently large universal constant $C > 0$, for any $(\mathbf{x},y) \notin \mathbf{Z}$.

Therefore, in total, $\widehat{\mathbb{P}}[\mathbb{P}[h(\mathbf{x}) \neq y|(\mathbf{x},y) \notin \mathbf{Z}] > \eta] \leq \frac{1}{\eta}\exp(-T/C)$. We set $T = C\ln\left(\frac{1}{\epsilon\delta}\right) \geq C\ln\left(\frac{1}{\eta\delta}\right)$ to obtain $\widehat{\mathbb{P}}[\mathbb{P}[h(\mathbf{x}) \neq y|(\mathbf{x},y) \notin \mathbf{Z}] > \eta] \leq \delta$ and, hence, with probability at least $1-\delta$ over the randomness of h we overall have $\mathbb{P}[h(\mathbf{x}) \neq y] \leq 4\eta$.

Completeness. Completeness follows by a standard Hoeffding bound. □

3.14 Auxiliary Propositions

Let $\mathcal{N}(0, I_d)$ denote the standard multivariate Gaussian distribution over $\mathbb{R}^d$ and $\mathrm{Unif}(\{\pm 1\}^d)$ denote the uniform distribution over the hypercube $\{\pm 1\}^d$. For each of these distributions, we show that the sandwiching polynomials of any binary concept have coefficients that are absolutely bounded, that the empirical moments concentrate around the true ones and that the empirical squared error of polynomials with bounded degree and coefficients uniformly converges to the true squared error. These properties are used in order to apply Theorem 16 to obtain TDS learning algorithms for a number of classes under the Gaussian and Uniform distributions.

3.14.1 Properties of Gaussian Distribution

We prove the following fact about the Gaussian distribution.

Lemma 20 (Properties of the Gaussian Distribution)**.** *Let D be the standard Gaussian $\mathcal{N}(0, I_d)$ over $\mathbb{R}^d$. Then the following are true.*

(*i*) *(Coefficient Bound) Suppose that for some $\epsilon \in (0,1]$, $k > 0$ and some concept class $\mathcal{C} \subseteq \mathbb{R}^d \to \{\pm 1\}$, the ϵ-approximate L_2 sandwiching degree of $\mathcal{C}$ w.r.t. $\mathcal{N}(0, I_d)$ is at most k. Then, the coefficients of the sandwiching polynomials for $\mathcal{C}$ are bounded by $B = O(d)^k$.*

(*ii*) *(Concentration) For any $\delta \in (0,1), \Delta > 0$ and $k > 0$, if X is a set of independent samples from D with size at least $m_{\mathrm{conc}} = \frac{O(dk)^k}{\Delta^2 \cdot \delta}$ then, with probability at least $1-\delta$ over the randomness of X, for any $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq k$ it holds $|\mathbb{E}_X[\mathbf{x}^\alpha] - \mathbb{E}_D[\mathbf{x}^\alpha]| \leq \Delta$.*

(*iii*) *(Generalization) For any $\epsilon > 0$, $\delta \in (0,1), B > 0$, $k > 0$, and any distribution $\mathcal{D}_{\mathcal{XY}}$ over $\mathbb{R}^d \times \{\pm 1\}$ whose marginal on $\mathbb{R}^d$ is D, if S is a set of independent samples from $\mathcal{D}_{\mathcal{XY}}$ with*

size at least $m_{\mathrm{gen}} = \tilde{O}(\frac{B^8}{\epsilon^4\delta}) \cdot d^{O(k)}$ *then, with probability at least* $1-\delta$ *over the randomness of* S*, for any polynomial* p *of degree at most* k *and coefficients that are absolutely bounded by* B *it holds* $|\mathbb{E}_S[(y-p(\mathbf{x}))^2] - \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2]| \leq \epsilon$.

Proof. We will prove each part of the Lemma separately.

Part (*i*). Suppose that $p_{\mathrm{up}}, p_{\mathrm{down}}$ are 1-sandwiching polynomials for some concept $f \in \mathcal{C}$ with degree at most k. Then, we have the following.

$$\begin{aligned}\|p_{\mathrm{down}}\|_{L_2(D)} &\leq \|p_{\mathrm{up}} - f\|_{L_2(D)} + \|f\|_{L_2 D} \\ &\leq \|p_{\mathrm{up}} - p_{\mathrm{down}}\|_2 + 1 \leq 2\end{aligned}$$

Since D is the standard Gaussian distribution, the quantity $\|p_{\mathrm{down}}\|^2_{L_2(D)}$ equals to the sum of the squares of the coefficients of the Hermite expansion of p_{down} (see e.g. [O'D14]). Therefore, each Hermite coefficient of p_{down} is absolutely bounded by 2. Each Hermite polynomial of degree at most k has coefficients that are absolutely bounded by 2^k. Since p_{down} has degree at most k, each coefficient of p_{down} is absolutely bounded by $d^{O(k)}$.

Part (*ii*). Suppose that $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq k$. Then, the worst case regarding moment concentration is $\alpha_1 = k$. For a sample X from D, we apply Chebyshev's inequality on the random variable $z = |\mathbb{E}_X[\mathbf{x}_1^k] - \mathbb{E}_D[\mathbf{x}_1^k]|$ and by bounding $\mathbb{E}[z^2]$ by $\mathbb{E}_D[\mathbf{x}_1^{2k}]$ we have that for any $\Delta > 0$, $z \leq \Delta$ with probability at least $1 - \frac{(Ck)^k}{|X|\Delta^2}$, where the randomness is over the random choice of X and $C > 0$ is a sufficiently large universal constant (for bounds on the Gaussian moments, see, e.g., Proposition 2.5.2 in [Ver18]). Since we need the result to hold for all α simultaneously, the result follows by a union bound.

Part (*iii*). We define $\mathcal{P}$ to be the class of polynomials over $\mathbb{R}^d$ with degree at most k and coefficients that are absolutely bounded by B. Let $T > 0$ to be disclosed later and $m = |S|$. We will first show that with probability at least $1 - \delta/2$ over the choice of S, we have

$$\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2] \leq \mathbb{E}_S[(y-p(\mathbf{x}))^2] + \epsilon \text{ for all } p \in \mathcal{P}$$

We aim to apply some standard uniform convergence argument, but in order to do so we first need to ensure certain boundedness conditions as follows.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2] =& \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2 \cdot \mathbb{1}\{\forall q \in \mathcal{P} : |q(\mathbf{x})| \leq T\}] \\ &+ \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2 \cdot \mathbb{1}\{\exists q \in \mathcal{P} : |q(\mathbf{x})| > T\}]\end{aligned}$$

where we have $\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2 \cdot \mathbb{1}\{\forall q \in \mathcal{P} : |q(\mathbf{x})| \leq T\}] \leq \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2 \mid \forall q \in \mathcal{P} : |q(\mathbf{x})| \leq T]$. Let $\mathcal{D}'_{\mathcal{XY}}$ be the distribution that corresponds to $\mathcal{D}_{\mathcal{XY}}$ conditioned on the event $\{\forall q \in \mathcal{P} : |q(\mathbf{x})| \leq T\}$ and let $S' = \{(\mathbf{x}, y) \in S : |q(\mathbf{x})| \leq T, \forall q \in \mathcal{P}\}$. By standard arguments using Rademacher complexity bounds for bounded losses (see, e.g., Theorems 5.5 and 10.3 in [MRT18]) we have that for some sufficiently large universal constant $C > 0$, with probability at

least $1 - \delta/4$, we have for any $p \in \mathcal{P}$

$$\mathbb{E}_{\mathcal{D}'_{\mathcal{X}\mathcal{Y}}}[(y - p(\mathbf{x}))^2] \leq \mathbb{E}_{S'}[(y - p(\mathbf{x}))^2] + T^4 \cdot \frac{B + \sqrt{\log(1/\delta)}}{\sqrt{m/C}} \tag{3.44}$$

We now need to link $\mathbb{E}_{S'}[(y - p(\mathbf{x}))^2]$ to $\mathbb{E}_S[(y - p(\mathbf{x}))^2]$. We have the following.

$$\begin{aligned}
\mathbb{E}_S[(y - p(\mathbf{x}))^2] &\geq (1 - \mathop{\mathbb{P}}_S[\exists q \in \mathcal{P} : |q(\mathbf{x})| > T])\mathbb{E}_{S'}[(y - p(\mathbf{x}))^2] \\
&\geq \mathbb{E}_{S'}[(y - p(\mathbf{x}))^2] - \mathop{\mathbb{P}}_S[\exists q \in \mathcal{P} : |q(\mathbf{x})| > T] \cdot 2T^2 \\
&\qquad \text{(since } y \in \{\pm 1\} \text{ and } p \in \mathcal{P}\text{)}
\end{aligned}$$

We will upper bound the quantity $\mathbb{P}_S[\exists q \in \mathcal{P} : |q(\mathbf{x})| > T]$. We have

$$\begin{aligned}
\mathop{\mathbb{P}}_S[\exists q \in \mathcal{P} : |q(\mathbf{x})| > T] &= \mathop{\mathbb{P}}_S\Big[\exists (q_\alpha)_{\|\alpha\|_1 \leq k} \in [-B, B]^{d^k} : \Big|\sum_\alpha q_\alpha \mathbf{x}^\alpha\Big| > T\Big] \\
&\leq \sum_{\alpha : \|\alpha\|_1 \leq k} \mathop{\mathbb{P}}_S\Big[|\mathbf{x}^\alpha| \geq \frac{T}{Bd^k}\Big] \\
&\leq \sum_{\alpha : \|\alpha\|_1 \leq k} \mathop{\mathbb{P}}_D\Big[|\mathbf{x}^\alpha| \geq \frac{T}{Bd^k}\Big] + \frac{d^k}{\sqrt{2m}} \log\Big(\frac{8}{\delta}\Big), \text{ w.p. at least } 1 - \delta/4
\end{aligned} \tag{3.45}$$

In the last step, we used a standard Chernoff-Hoeffding bound. We now bound $\sum_{\alpha : \|\alpha\|_1 \leq k} \mathbb{P}_D[|\mathbf{x}^\alpha| \geq \frac{T}{Bd^k}]$. Recall that $D = \mathcal{N}(0, I_d)$ and therefore the worst case for α regarding concentration is the case $\alpha_1 = k$. We therefore obtain the following via Gaussian concentration.

$$\begin{aligned}
\sum_{\alpha : \|\alpha\|_1 \leq k} \mathop{\mathbb{P}}_D\Big[|\mathbf{x}^\alpha| \geq \frac{T}{Bd^k}\Big] &\leq d^k \mathop{\mathbb{P}}_D\Big[|\mathbf{x}_1^k| \geq \frac{T}{Bd^k}\Big] \\
&\leq d^k \exp\Big(-\frac{1}{2} \cdot \frac{T^{1/k}}{B^{1/k} d}\Big)
\end{aligned} \tag{3.46}$$

It remains to bound the term $\mathbb{E}_{\mathcal{D}_{\mathcal{X}\mathcal{Y}}}[(y - p(\mathbf{x}))^2 \cdot \mathbb{1}\{\exists q \in \mathcal{P} : |q(\mathbf{x})| > T\}]$. By applying the Cauchy-Schwarz inequality, it is sufficient to bound $\sqrt{\mathbb{E}_{\mathcal{D}_{\mathcal{X}\mathcal{Y}}}[(y - p(\mathbf{x}))^4]} \cdot \sqrt{\mathbb{P}_D[\exists q \in \mathcal{P} : |q(\mathbf{x})| > T]}$. For the second term, we use Equation (3.46). For the first term, we have the following for some sufficiently large constant $C > 0$.

$$\begin{aligned}\mathbb{E}_D[(y-p(\mathbf{x}))^4] &\leq 8 + 8\mathbb{E}_D[p^4(\mathbf{x})] \\ &\leq 8 + B^4 d^{4k} \sum_{\|\alpha\|_1 \leq 4k} \prod_{i:\alpha_i>0} \mathbb{E}_D[\mathbf{x}_i^{\alpha_i}] \quad \text{(since } \deg(p^4) \leq 4k \text{ and } |(p^4)_\alpha| \leq B^4 d^{4k}\text{)} \\ &\leq B^4 d^{8k} (Ck)^{2k} \qquad \text{(since } D = \mathcal{N}(0, I_d)\text{, see Proposition 2.5.2 in [Ver18])}\end{aligned}$$

Using the above inequality along with (3.44), (3.45) and (3.46) we obtain that $\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y - p(\mathbf{x}))^2] - \mathbb{E}_S[(y - p(\mathbf{x}))^2]$ is upper bounded by the following quantity for some sufficiently large universal constant $C > 0$

$$\begin{aligned}&T^4 \cdot \frac{B + \sqrt{\log(1/\delta)}}{\sqrt{m/C}} + 2T^2 d^k \exp\left(-\frac{1}{2} \cdot \left(\frac{T}{Bd^k}\right)^{1/k}\right) + \\ &+ 2T^2 \frac{d^k}{\sqrt{2m}} \log\Big(\frac{10}{\delta}\Big) + B^2 d^{4k} (Ck)^k d^{k/2} \exp\left(-\frac{1}{4} \cdot \left(\frac{T}{Bd^k}\right)^{1/k}\right),\end{aligned}$$

which is at most ϵ when we choose m, T as follows for some universal constant $C > 0$ (possibly larger than the previously defined constants for which we used the same letter) for the choice $T = CB(4d)^k k \log\big(\frac{Bdk}{\epsilon}\big)$ and $m = \frac{C}{\epsilon^2}(B^2 + \log\big(\frac{1}{\delta}\big))B^8(4d)^{8k}k^8 \log\big(\frac{Bdk}{\epsilon}\big) = \tilde{O}(\frac{B}{\epsilon^2}) \cdot O(d)^{8k} \cdot \log(1/\delta)$.

In order to bound the symmetric difference, we also need to bound the quantity $\mathbb{E}_S[(y - p(\mathbf{x}))^2] - \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y - p(\mathbf{x}))^2]$, which we may do following a similar reasoning, but requiring, at times, bounds on quantities that correspond to empirical expectations (instead of expectations over the population distribution). In particular, we will require a bound on $\mathbb{E}_S[(y - p(\mathbf{x}))^4]$, which can be reduced to bounding $\mathbb{E}_S[p^4(\mathbf{x})]$, for which we may use part (ii), demanding $m \geq d^{O(k)}/\delta$ to obtain

$$\mathbb{E}_S[p^4(\mathbf{x})] \leq 2B^4 d^{4k} (Ck)^k$$

Overall, this step will introduce the additional requirement that $m \geq \frac{B^8}{\epsilon^4 \delta} d^{16k} (Ck)^{4k} \log^2(\frac{1}{\delta})$. Therefore, overall, for $m \geq m_{\text{gen}} = \tilde{O}(\frac{B^8}{\epsilon^2 \delta}) \cdot d^{O(k)} \cdot \log^2(\frac{1}{\delta})$, we have the desired result. □

3.14.2 Properties of Uniform Distribution

We prove the following fact about the uniform distribution.

Lemma 21 (Properties of the Uniform Distribution)**.** *Let D be the uniform distribution over the hypercube $\mathrm{Unif}(\{\pm 1\}^d)$ and $C > 0$ some sufficiently large constant. Then the following are true.*

(*i*) *(Coefficient Bound) Suppose that for some $\epsilon \in (0, 1]$, $k > 0$ and some concept class $\mathcal{C} \subseteq \mathbb{R}^d \to \{\pm 1\}$, the ϵ-approximate L_2 sandwiching degree of $\mathcal{C}$ w.r.t. D is at most k. Then, the coefficients of the sandwiching polynomials for $\mathcal{C}$ are absolutely bounded by $B = 2$.*

(*ii*) *(Concentration) For any $\delta \in (0,1), \Delta > 0$ and $k > 0$, if X is a set of independent samples from D with size at least $m_{\text{conc}} = \frac{Ck}{\Delta^2}\log\left(\frac{d}{\delta}\right)$ then, with probability at least $1-\delta$ over the randomness of X, for any $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq k$ it holds $|\mathbb{E}_X[\mathbf{x}^\alpha] - \mathbb{E}_D[\mathbf{x}^\alpha]| \leq \Delta$.*

(*iii*) *(Generalization) For any $\epsilon > 0$, $\delta \in (0,1), B > 0$, $k > 0$, and any distribution $\mathcal{D}_{\mathcal{XY}}$ over $\mathbb{R}^d \times \{\pm 1\}$ whose marginal on $\mathbb{R}^d$ is D, if S is a set of independent samples from $\mathcal{D}_{\mathcal{XY}}$ with size at least $m_{\text{gen}} = \tilde{O}(\frac{1}{\epsilon^2}) \cdot B^{O(1)} \cdot d^{O(k)} \cdot \log\left(\frac{1}{\delta}\right)$ then, with probability at least $1-\delta$ over the randomness of S, we have that for any polynomial p of degree at most k and coefficients that are absolutely bounded by B it holds $|\mathbb{E}_S[(y-p(\mathbf{x}))^2] - \mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2]| \leq \epsilon$.*

Proof. We will prove each part of the Lemma separately.

Part (*i*). Suppose that $p_{\text{up}}, p_{\text{down}}$ are 1-sandwiching polynomials for some concept $f \in \mathcal{C}$ with degree at most k. Then, we have the following.

$$\begin{aligned}\|p_{\text{down}}\|_{L_2(D)} &\leq \|p_{\text{up}} - f\|_{L_2(D)} + \|f\|_{L_2 D} \\ &\leq \|p_{\text{up}} - p_{\text{down}}\|_2 + 1 \leq 2\end{aligned}$$

Since D is the uniform distribution, the quantity $\|p_{\text{down}}\|^2_{L_2(D)}$ equals to the sum of the squares of the coefficients of p_{down} (see e.g. [O'D14]). Therefore, each coefficient of p_{down} is absolutely bounded by 2.

Part (*ii*). Suppose that $\alpha \in \{0,1\}^d$ with $\|\alpha\|_1 \leq k$. For a sample X from D, we apply Hoeffding's inequality on the random variable $z = |\mathbb{E}_X[\mathbf{x}^\alpha] - \mathbb{E}_D[\mathbf{x}^\alpha]|$ and by observing that $\mathbf{x}^\alpha \in \{\pm 1\}$ we have that the probability that $z > \Delta$ is at most $2\exp(-|X|\Delta^2/10)$. We obtain the desired result by a union bound.

Part (*iii*). We define $\mathcal{P}$ to be the class of polynomials over $\{\pm 1\}^d$ with degree at most k and coefficients that are absolutely bounded by B. Let $T > 0$ to be disclosed later and $m = |S|$. We will show that with probability at least $1-\delta$ over the choice of S, we have

$$|\mathbb{E}_{\mathcal{D}_{\mathcal{XY}}}[(y-p(\mathbf{x}))^2] - \mathbb{E}_S[(y-p(\mathbf{x}))^2]| \leq \epsilon \text{ for all } p \in \mathcal{P}$$

We apply some standard uniform convergence argument, by observing that $(y-p(\mathbf{x}))^2 \leq 2 + 2B^2 d^k$. In particular by standard arguments using Rademacher complexity bounds for bounded losses (see, e.g., Theorems 5.5 and 10.3 in [MRT18]) we obtain the desired result. □

Part II

Closing Computational-to-Statistical Gaps via Sublinear-Time Correction

Chapter 4

Properly learning monotone functions via local correction

4.1 Chapter Overview.

Proper learning of monotone functions. Consider the *proper learning* problem for monotone functions:

Given i.i.d uniform labeled examples from an unknown monotone $g : \{0,1\}^d \to \{0,1\}$, output a monotone, ϵ-accurate predictor $\widehat{g} : \{0,1\}^d \to \{0,1\}$ — that is, a circuit computing a monotone function that agrees with g on at least $1-\epsilon$ fraction of the domain.

For over 25 years there has been a large statistical-to-computational gap in our understanding of this problem. A $2^{\tilde{O}(\sqrt{d}/\epsilon)}$-time *improper* learning algorithm — that is, an algorithm that outputs a predictor f that is accurate but not guaranteed to be monotone — is given in [BT96]. One could use the output of this algorithm to obtain a monotone $\widehat{g}$ by computing f on every element of $\{0,1\}^d$ and solving a linear program to obtain the closest monotone function to f. Although this gives a $2^{\tilde{O}(\sqrt{d}/\epsilon)}$-sample algorithm, the run-time is $2^{\Omega(d)}$. No proper learning algorithm with faster run-time (i.e. $2^{o(d)}$) was known, even given query access to g.

Through a new connection with local computation algorithms, we close this gap by giving a $2^{\tilde{O}(\sqrt{d}/\epsilon)}$-time algorithm for this problem. Note that the running time essentially matches that of the aforementioned improper learning algorithm of [BT96]. Moreover, our algorithm is essentially optimal, due to the $2^{\tilde{\Omega}(\sqrt{d})}$ query lower bound of [BCO$^+$15]. Furthermore, our algorithm is robust to adversarial noise in the labels. Specifically, in the agnostic learning model of Kearns, Schapire, and Sellie [KSS94b], our algorithm can handle a noise rate of $\Omega(\epsilon)$.

Monotonicity testing. The question of testing monotonicity of an unknown Boolean function over $\{0,1\}^d$ (given query access) has received a large amount of attention [GGL$^+$00, DGL$^+$99, CS13a, CST14, CDST15, KMS15, BB21, CWX17]. However, the algorithms in this line of work

possess the drawback of having a large *tolerance ratio*[1], i.e. they will reject some functions that are extremely close to monotone. The more recent work of [PRW22] gives a tester with tolerance ratio of $\tilde{O}(\sqrt{d})$, and this is the best tolerance ratio known[2] for a $2^{o(d)}$-run-time algorithm.

As a simple corollary of our proper learning algorithm, one can already achieve *constant* tolerance ratio for monotonicity testing with $2^{\tilde{O}\left(\sqrt{d}/\epsilon\right)}$ run-time via a well-known connection between learning and testing [GGR98]. (We emphasize that, to draw this corollary, it is critical that the learning algorithm is proper and robust to noise in labels.) Even more, building more directly on our technical ideas, we present a constant-tolerance monotonicity tester with *exponentially better dependence on ϵ of* $2^{\tilde{O}\left(\sqrt{d \log \frac{1}{\epsilon}}\right)}$. This approach also yields a tolerant monotonicity tester for functions over general posets, which we describe in detail in Corollary 9. By a standard reduction [PRR04], this also gives an approximator for distance to monotonicity (within a constant multiplicative error plus an ϵN additive error) in functions over general posets as well.

4.1.1 Monotonicity correction via sorting on partially ordered sets.

The poset-sorting problem. One of the core ideas in this chapter is to create a new bridge between our learning task and recent exciting developments on parallelizing greedy algorithms [Gha15, GU19, Gha22]. These developments accomplish speedups for classic graph problems in the setting of local and distributed computing. They build on a large body of work in local computation algorithms and distributed graph algorithms: some examples include [RTVX11a, ARVX12a, LRY17, GHL+15, RV16, EMR14, CFG+19, ELMR21, PRVY19, AL21, BGR21, GR21]. In order to accomplish this, we introduce the problem of poset sorting, that is, sorting binary values on a partially ordered set (poset), which we believe is of independent interest:

Let P be a poset of N elements with longest chain length at most h and such that every element has at most Δ predecessors or successors. Given a binary labeling f of elements of P, output a new binary labeling f_{mon} that (i) is monotone with respect to the partial order in P (ii) can be obtained from f by a sequence of swaps of monotonicity-violating label pairs.

Clearly, there is a greedy algorithm for this task that keeps swapping monotonicity-violating pairs of labels until there are none left. The challenge is to do this in a distributed fashion. Among the many distributed and local computation models, the one that turns out relevant to us is the *local computation algorithm* (LCA) model, defined formally in Section 4.2. In brief, to be an LCA, an algorithm should be local in the sense that it should not need to read the entire assignment of labels to vertices in order to determine which label will end up at some particular vertex x. It should suffice to read only the labels of vertices that lie in a restricted neighborhood of x — ideally not much more than the set of vertices one would need to read in order to determine whether x's own label violates monotonicity. We explain the algorithm itself in Subsection 4.1.2.

[1] A tester for monotonicity distinguishes a monotone function from a function that is ϵ-far from monotone. In this chapter, we use tolerance ratio α to mean that the tester will accept a function ϵ/α-close to monotone.

[2] Note that Theorem 1.8 of [CGG+17] gives a $2^{o(d)}$-query algorithm (no run-time bound is claimed). The run-time is still $2^{\Omega(d)}$ due to step 8 of their Algorithm 2 on page 31.

Proper learning via local correction. Why is an algorithm for sorting on a poset P relevant to proper learning? Suppose we obtained an improper predictor f via the algorithm of [BT96], in the form of a small circuit computing f. Then, let monotone f_{mon} be obtained from f by flipping a sequence of monotonicity-violating labels. It is not hard to argue that since f is $O(\epsilon)$-close to the monotone g we are trying to learn, then so is any f_{mon} obtained in this manner (see Proposition 45 for more details). What the LCA allows us to do is to transform the circuit computing f into a *small* circuit computing such f_{mon}. The reason is that to evaluate f_{mon} at a given element x the LCA evaluates f on only a small number of points and has an appropriately fast run-time. This allows us to augment the circuit for f with a circuit that executes this sorting algorithm, and as a result obtain a small circuit for f_{mon}.

In other words, an LCA for sorting on a poset is a *local corrector* for monotonicity: an algorithm that takes some input x, makes queries to a black-box function f, and outputs $f_{mon}(x)$ where f_{mon} is a monotone function that is close to f in Hamming distance, if such a function exists. Examples of local correctors for various function properties can be found in . A local corrector, combined with any improper hypothesis, yields a proper hypothesis. The efficiency of the LCA determines how quickly the proper hypothesis can be evaluated.

4.1.2 Our LCA for sorting on a poset.

When the longest chain length h is 1, then the poset-sorting problem is equivalent to the classical problem of finding a maximal matching on a bipartite graph with N vertices and maximum degree at most Δ. We note that a recent LCA by Ghaffari [Gha22] handles this problem using a run-time of only $\text{poly}(\Delta, \log(N/\delta))$.

For larger values of h, a naive approach would be to execute a sequence of phases, in each of which a maximal matching between monotonicity-violating labels is produced and the labels that are matched with each other are swapped. One can show that $O(h)$ such phases suffice and sometimes necessary if the matchings are arbitrary. If the algorithm by Ghaffari [Gha22] is used to implement each of the phases, this yields a run-time of $(\Delta \log(N/\delta))^{\Theta(h)}$. For properly learning monotone functions over the Boolean cube, the parameters we are primarily interested in are $h = \Theta(\sqrt{d})$, $\Delta = 2^{\tilde{\Theta}(\sqrt{d})}$ and $N = \Theta(2^d)$. (A slight subtlety in our argument is that we need to work over truncated hypercube, i.e. handle separately $O(\epsilon)$ fraction of the elements with too high or too low Hamming weight, which is a standard technique). The naive approach then gives us a run-time of $(\Delta \log(N/\delta))^{\Theta(h)} = 2^{\Omega(d)}$, which is too slow for our purposes.

We beat the naive approach by enforcing that the maximal matchings at each step only include pairs of vertices that are sufficiently far away in the graph. We show that after each matching step, the greatest distance between pairs that violate monotonicity reduces by a factor of 2. This allows us to reduce the exponent from h to $\log h$, which is sufficiently fast to yield an essentially optimal proper learning algorithm for monotone functions.

4.1.3 Other related work

The problem of locally correcting monotonicity has been studied in [ACSL08, ACSL07, SS10a, BGJ$^+$12, AJMR14] in various parameter settings. The work of [ACSL08] introduces the problem of online property reconstruction and gives an algorithm for correcting monotonicity for real-valued functions over the discrete number line. The work of [PRR04] gives a tolerant tester in the same setting. The work of [SS10a] introduces the framework of *local* property reconstruction, which is the same framework our approach uses (i.e. local correction by memoryless LCA). They give a local corrector for functions over the hypergrid $[d]^d$, with large dependence on the dimension d but small dependence on d. Lower bounds for monotonicity correction in other error regimes are given in [BGJ$^+$12, AJMR14]. The problem of approximating the distance to monotonicity, which is strongly related to tolerant testing, has been studied in [PRW22, ACSL07, PRR04].

A proper learning algorithm for a function class that generalizes monotone functions is given in [CGG$^+$17]. Proper learning of restricted classes of monotone functions has been studied in [JLSW11, Ang88, YBC13, JLSW11, BLQT22]. The question of weak learning of monotone functions has also received attention [KV89, BT96, BBL98, AM06, OW09]. The latter line of work investigates proper learning algorithms that have very fast run-time at the cost of having accuracy of only $\frac{1}{2} + \frac{1}{\mathrm{poly}(d)}$.

In addition to the Boolean cube, testing monotonicity has also been studied on hypergrids, see for example [CS13b], [BRY14b] [CS13c], [BCS18], [BCS20]. Also see [CS19] for monotonicity testing of functions with bounded influence.

4.1.4 Organization of this chapter

In Section 4.2.3 we define the LCA model and state the maximal matching result of [Gha22]. In Section 4.3 we state and prove the main proper learning and testing results as consequences of our local poset sorting algorithm. In Section 4.4 we present pseudocode for the local sorting algorithm and analyze its correctness and complexity.

4.2 Preliminaries

4.2.1 Notation (posets and distances)

Let x and y be elements of a poset P.[3] We use $\preceq$ to denote the ordering relation on P. We say $x \prec y$ ("x is a predecessor of y") if $x \preceq y$ and $x \neq y$. Also, we say $x \succeq y$ if $y \preceq x$, and $x \succ y$ ("x is a successor of y") if $y \prec x$. We say x and y are *incomparable* if neither $x \prec y$ nor $x \succ y$ holds. We say that x is an *immediate predecessor* of y if $x \prec y$ and there is no z in P for which $x \prec z \prec y$. The notion of an *immediate successor* is defined analogously.

The *graph* of a poset P (a.k.a. *Hasse diagram* of P) is a directed graph, in which elements of P are the vertices, and there is an edge from x to y whenever x is an immediate predecessor of y.

[3] In this chapter all posets are assumed to be finite.

Clearly, the graph of any poset is a DAG. Additionally, it is immediate that the poset itself is unambiguously determined by its graph, and we will refer to the poset and its graph interchangeably.

Definition 18 (Graph distance and height)**.** *We will write* $\mathrm{dist}(x, y)$ *to denote the length of the* ***longest*** *directed path*[4] *between* x *and* y *in the Hasse diagram of* P*. The* ***height*** *of a poset* P *is the length of the longest directed path between any two elements in* P*.*

Definition 19 (Function distance)**.** *For a pair of functions* $f_1, f_2 : P \to \{0, 1\}$*, the* ***distance*** $\|f_1 - f_2\|$ *is the fraction of elements* x *in* P *on which* $f_1(x) \neq f_2(x)$*.*

Definition 20 (Distance to monotonicity)**.** *For a function* $f : P \to \{0, 1\}$*, the* ***distance to monotonicity*** *is defined as* $\min_{\text{monotone Boolean } f'} (\|f - f'\|)$*.*

4.2.2 Agnostic learning setting

Now, we formally describe the setting of agnostic learning under the uniform distribution. The learning algorithm is given i.i.d. example-label pairs from some distribution over $\{0, 1\}^d \times \{0, 1\}$, where the marginal distribution over examples is uniform. The *generalization error* (which we also refer to simply as *error*) of a predictor is the probability it misclassifies a fresh example-label pair. The goal of agnostic learning is to produce a predictor $\widehat{g}$ with good generalization error. Guarantees are produced in terms of the lowest generalization error among all hypotheses in the function class (in our case monotone functions), which we denote as opt.

4.2.3 The LCA model

The Local Computation Algorithm (LCA) model captures the ability to provide query access to parts of an output in sublinear time. In this chapter, we use the LCA model for the problems of maximal matching and poset sorting. An LCA is given access to a random bit-string and to the input: for example, in the case of maximal matching, this input is the adjacency list of the graph. Upon receiving an edge query e in G, the LCA should respond "yes" or "no." Responses to different edges must be consistent with a single legal maximal matching. Similarly, for poset sorting on poset P, a query to the LCA is any element $x \in P$, and the LCA, which is provided with access to function $f : P \to \{0, 1\}$, must respond with $f_{\text{mon}}(x)$ so that the function f_{mon} is monotone. Responses to different inputs must be consistent with a single legal sorting of f. In both cases, the LCA's answers may depend on a random bit-string[5] that persists between queries, but otherwise, responses must be[6] *memoryless* – i.e., they cannot depend on previous queries to or responses from the LCA.

[4] If x and y are incomparable, then $\mathrm{dist}(x, y)$ is undefined.

[5] In all cases we consider, the random bit-string is short enough for the LCA to read as a whole (in contrast, in some work exponentially long bit-strings are considered).

[6] Sometimes LCAs are considered that are not memoryless, but in this chapter whenever we refer to an LCA, we imply it is memoryless.

In terms of performance, we want the following quantities to be as small as possible (i) the *query complexity*, i.e. the number of probes to the input the LCA makes to respond to a single query, (ii) the *run-time* the LCA needs to respond to a query, (iii) the length of the random bit-string used by the LCA, (iv) the probability over the random bit-string that the LCA fails to satisfy the problem specifications.

We will use a recent powerful result[7] of Ghaffari [Gha22], which gives an efficient algorithm for answering membership queries to a maximal independent set. (We emphasize that *maximal* independent set is defined to be an independent set that cannot be made into a larger independent set by adding an extra vertex, and *maximal matching* is defined analogously.) We note that in an earlier version of this Chapter (which was written before [Gha22] was available) we used the theorem of Ghaffari and Uitto [GU19] for this purpose.

Theorem 26 ([Gha22])**.** *There is an LCA that takes all-neighbor*[8] *access to a graph G with N vertices and largest degree at most Δ, and gives membership access to a* maximal independent set *on the graph. The query complexity and the run-time of the LCA are $\mathrm{poly}(\Delta, \log(N/\delta))$, the length of the random bit-string is also $\mathrm{poly}(\Delta, \log(N/\delta))$ and the failure probability is at most δ.*

Corollary 7. *There is an LCA that takes all-neighbor access to a graph G, with N vertices and largest degree at most Δ, and gives membership access to a* maximal matching *on the graph. The query complexity and the run-time of the LCA is $\mathrm{poly}(\Delta, \log(N/\delta))$, the length of the random bit-string is also $\mathrm{poly}(\Delta, \log(N/\delta))$ and the failure probability is at most δ.*

Proof. This reduction is standard; see Section 4.5.1 for details. □

4.2.4 Boolean hypercube.

Definition 21. *The d-*dimensional Boolean hypercube *is the set $\{0,1\}^d$. For $x, y \in \{0,1\}^d$, we say $x \preceq y$ if for all $i \in \{1, \cdots, d\}$ one has $x_i \leq y_i$. It is immediate that $\{0,1\}^d$ is a poset with 2^d elements.*

We also define the truncated hypercube

$$H_\epsilon^d := \left\{ x \in \{0,1\}^d : \left| \sum_i x_i - \frac{d}{2} \right| \leq \sqrt{\frac{d}{2} \log \frac{2}{\epsilon}} \right\},$$

[7] We need to comment on some superficial differences between Theorem 26 and the main theorem of [Gha22], which does not mention run-time and bit-string length explicitly and also has $\delta = \mathrm{poly}\left(\frac{1}{N}\right)$. The following observations are not novel in any way, and some of them are alluded to in [GU19, Gha22], but we explain them here for completeness. The bound on run-time follows by direct inspection of their algorithm. The length of the bit-string can be reduced to $\mathrm{poly}(\Delta, \log(N/\delta))$ via the standard method [ARVX12a, LRY17] of replacing i.i.d. random bits with k-wise independent random bits for k large enough ($k = \mathrm{poly}(\Delta, \log(N/\delta))$ suffices, as number of those i.i.d. random bits accessed per query is $\mathrm{poly}(\Delta, \log(N/\delta))$). Although, the failure probability bound δ in the original theorem of [Gha22] is set to be $\mathrm{poly}\left(\frac{1}{N}\right)$, it can be boosted to arbitrary δ by adding extra disconnected vertices to our graph until this value reaches δ for the new number of vertices N'. Query access to this new graph can be simulated via query access to the original graph with inconsequential overhead. Overall, this costs one an extra $\mathrm{polylog}(1/\delta)$ factor in query complexity and run-time.

[8] I.e. when queried a vertex v, the oracle returns all the neighbors of v.

Via Hoeffding's bound, we have that the fraction of elements in $\{0,1\}^d$ *that are not also in* H_n^ϵ *is at most* $2\exp\left(-\frac{2t^2}{d}\right) = \epsilon$.

Known results about learning monotone functions over Boolean hypercube.

Theorem 27 (Learnability of monotone functions [BT96])**.** *There is an algorithm that, for any monotone function* $g : \{0,1\}^d \to \{0,1\}$*, given i.i.d. example-label pairs* $(x_i, g(x_i))$*, with* x_i *uniform in* $\{0,1\}^d$*, returns a circuit computing a predictor* $\widehat{g}$*, such that* $\|g - \widehat{g}\| \leq \epsilon$*. The algorithm uses* $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)} \log\left(\frac{1}{\delta}\right)$ *samples and run-time, where* δ *is the failure probability bound.*

The theorem below follows via low-degree concentration result of [BT96], Remark 4 on page 6 of [KKMS08] a refinement by [FKV17] and standard failure probability reduction via repetition:

Theorem 28 (Agnostic learnability of monotone functions)**.** *In the agnostic setting with examples distributed uniformly on* $\{0,1\}^d$*, there is an algorithm that returns a circuit with generalization error at most* opt $+ \epsilon$*, where* opt *is the error of the best monotone function. The algorithm uses* $d^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)} \log\left(\frac{1}{\delta}\right)$ *samples and run-time, where* δ *is the failure probability bound.*

4.3 Main result and consequences

We first present the formal statement for our LCA for the poset sorting problem, from which every other result in this section is derived. The algorithm and analysis are presented in Section 4.4.

Theorem 29. *Let* P *be a poset of* N *elements and height* h*, such that each element in* P *has at most* Δ *predecessors or successors. Suppose we are given query access to the graph of* P*, i.e. for any* $x \in P$ *we can obtain the immediate predecessors or successors of* x*. Also, suppose we are given query access to some Boolean function* f *over* P*. Then, there is an LCA that solves the poset-sorting problem for* f *over* P*: in other words, it provides query access to a monotone function* $f_{mon} : P \to \{0,1\}$ *that can be obtained from* f *by a sequence of swaps of monotonicity-violating label pairs. The LCA has query complexity and run-time of* $\left(\Delta \log\left(\frac{N}{\delta}\right)\right)^{O(\log h)}$*, and it uses a random bit-string of length* $\mathrm{poly}(\Delta \log \frac{N}{\delta})$*. The failure probability of the LCA is at most* δ*.*

Remark 5. *In the setting of Theorem 29, suppose each element of* P *has at most* d *immediate predecessors or successors, and furthermore, that* P *is graded (all paths between* u *and* v *for any* u, v *have the same length). Then query complexity and the run-time of our LCA is also bounded by* $d^{O(h)}(\log(N/\delta))^{\log h}$*. The number of random bits used is at most* $d^{O(h)}\,\mathrm{polylog}(N/\delta)$*. In particular, when* P *is the truncated hypercube* H_ϵ^d *and* $\delta = 2^{-10n}$*, the query and time complexity are* $d^{O\left(\sqrt{d \log \frac{1}{\epsilon}}\right)}$.

The proof of Remark 5 is given in Section 4.5.4.

Proposition 45 (Local correction)**.** *In the setting of the previous problem, the distance between f and f_{mon} is at most twice the distance of f to monotonicity. Furthermore, the following extra property holds: for any monotone $q : P \to \{0,1\}$ we have $\|f_{mon} - q\| \leq \|f - q\|$.*

Proof. We first show the extra property. Recall that f_{mon} can be obtained from f via a sequence of swaps of monotonicity-violating labels. Since q is monotone, as a result of every single of this swaps the distance to q will either decrease or stay the same. Overall across all the swaps, this means that $\|f_{\text{mon}} - q\| \leq \|f - q\|$.

Taking q to be the closest monotone function to f and using the triangle inequality, we see that the distance between f and f_{mon} is at most twice the distance of f to monotonicity. □

The two corollaries about tolerant testing of monotone functions follow from our theorem above. We note that the success probability of $2/3$ can be improved via repetition to $1 - \delta$ at the cost of $\log\left(\frac{1}{\delta}\right)$ multiplicative factor in run-time and query complexity. We also note that the inverse tolerance ratio, given as 0.49, can be improved to any absolute constant less than 0.5.

Corollary 8 (Tolerant testing for the Boolean cube)**.** *Suppose we are given query access to an unknown Boolean function f. Then, there is an algorithm that uses $d^{O\left(\sqrt{d \log \frac{1}{\epsilon}}\right)}$ queries and run-time, and distinguishes whether the function f is 0.49ϵ-close or ϵ-far from monotone. The failure probability is at most $2/3$.*

Proof. We use the truncated hypercube $H^d_{0.005\epsilon}$ as our poset when using Theorem 29 (also using the refined run-time of Remark 5). This allows us to gain query access to a monotone f_{mon} on $H^d_{0.005\epsilon}$. Extend f_{mon} to all of $\{0,1\}^d$ by setting it to 1 above the upper truncation threshold and to 0 below the lower threshold. Clearly, f_{mon} is now also monotone on all of $\{0,1\}^d$.

We sample i.i.d. uniformly random elements of $\{0,1\}^d$, evaluate both f and f_{mon} on each these elements and obtain an estimate of $\|f - f_{\text{mon}}\|$ up to error 0.005ϵ. If f is ϵ-far from monotone, then the distance $\|f - f_{\text{mon}}\|$ is also at least ϵ, so the estimate will be at least 0.995ϵ. If f is 0.49ϵ-close to monotone, then there is some monotone function over $H^d_{0.005\epsilon}$ with which f disagrees on at most $0.49\epsilon \cdot 2^d$ elements of $H^d_{0.005\epsilon}$. The guarantee of Theorem 29 (via Proposition 45) tells us that then f and f_{mon} disagree on at most $0.98\epsilon \cdot 2^d$ elements of $H^d_{0.005\epsilon}$. Since there are only at most $0.005\epsilon \cdot 2^d$ elements in $\{0,1\}^d$ that are not in $H^d_{0.005\epsilon}$, we see that $\|f - f_{\text{mon}}\|$ is at most 0.985ϵ. Therefore, the estimate will be at most 0.99ϵ. Overall, checking if the estimate is greater than 0.992ϵ allows us to distinguish whether f is ϵ-far from monotone or 0.49ϵ-close to monotone.

For the estimation to succeed, we need to evaluate f and f_{mon} on $O\left(\frac{1}{\epsilon^2}\right)$ i.i.d. random elements of $\{0,1\}^d$. For the LCA of f_{mon} we can set the overall success probability parameter to be 0.1. A Chernoff bound and union bound argument then shows that overall success probability is at least $2/3$. For $H^d_{0.005\epsilon}$ our parameters are $h = O\left(\sqrt{d \log \frac{1}{\epsilon}}\right)$, $N = O(2^d)$ and each element of the poset has at most d immediate predecessors or successors. Overall, the run-time given by Remark 5 is $d^{O\left(\sqrt{d \log \frac{1}{\epsilon}}\right)}$.

□

Corollary 9 (Tolerant testing for general posets)**.** *Suppose we are in the setting of Theorem 29, and we also have access to an oracle giving us i.i.d. uniform elements in P. Then, there is an algorithm that uses $\Delta^{O(\log h \log\log \Delta)} (\log (N))^{O(\log h)} \frac{1}{\epsilon^2}$ queries and run-time, and distinguishes whether the function f is 0.49ϵ-close or ϵ-far from monotone. The failure probability is at most $2/3$.*

Proof. The proof is similar to the proof of Corollary 8, and is given in Section 4.5.2. □

Theorem 30 (Proper learnability of monotone functions)**.** *There is an algorithm that, for any monotone function $g : \{0,1\}^d \to \{0,1\}$, given i.i.d. example-label pairs $(x_i, g(x_i))$, with x_i uniform in $\{0,1\}^d$, returns a circuit computing a* ***monotone function*** *$\widehat{g}$, such that $\|g - \widehat{g}\| \leq \epsilon$. The algorithm uses $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$ samples and run-time and fails with probability at most $1/2^d$.*

Proof. The proper learner does the following:

1. Use the improper learner in Theorem 27 with error parameter $\frac{\epsilon}{10}$ and failure probability bound $1/2^{d+1}$. This gives a circuit computing a function f over $\{0,1\}^d$.
2. Obtain a circuit computing a function $f_{\text{mon}} : H^d_{\epsilon/10} \to \{0,1\}$ as follows. The circuit is computed via running the LCA from Theorem 29 with accuracy parameter equal to $\frac{\epsilon}{10}$ and failure probability bound equal to $1/2^{d+1}$, and with the oracle calls to a function replaced with an evaluation of the circuit f, restricted to $H^d_{\epsilon/10}$. The random bit-string used by the LCA is hard-coded into the circuit for h, so that the resulting circuit is deterministic.
3. Augment the circuit computing f_{mon} in order to extend this function into the whole of $\{0,1\}^d$ as follows. If $|x| > \frac{d}{2} + \sqrt{\frac{d}{2}\log\frac{20}{\epsilon}}$ then $f_{\text{mon}}(x) = 1$, and if $|x| < \frac{d}{2} - \sqrt{\frac{d}{2}\log\frac{20}{\epsilon}}$ then $f_{\text{mon}}(x) = 0$.
4. Output the circuit computing f_{mon}.

With probability at least $1/2^d$ both of the algorithms we invoke succeed, which we will assume henceforth.

Let us discuss the run-time. Step 1 runs in time $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$ and the circuit for g can therefore only have size at most $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$. For step 2, first observe that we have $\left|H^d_{\epsilon/10}\right| \leq 2^d$, $H^d_{\epsilon/10}$ has height $2\sqrt{\frac{d}{2}\log\frac{20}{\epsilon}}$ and each element in $H^d_{\epsilon/10}$ can have only at most d immediate predecessors and successors. Therefore, the LCA from Theorem 29 (refined via Remark 5) in this setting has run-time, query complexity and bit-string length of $d^{O(\sqrt{d}\log(1/\epsilon))}$. Since the circuit for f itself has size $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$, the overall run-time of the learning algorithm and the size of circuit computing f_{mon} is also $2^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$.

Finally, we argue correctness. Correctness of the LCA in Theorem 29 implies that f_{mon} is monotone over $H^d_{\epsilon/10}$ and we see that the extension of this function to $\{0,1\}^d$ in step 3 keeps it monotone.

Now, let g be the function we are trying to learn. Since we are in the realizable setting, g is monotone. Theorem 27 tells us that f and g disagree on at most $\frac{\epsilon}{10}2^d$ elements. This, together with Theorem 29, Remark 45 and the fact that g is monotone, tells us that f_{mon} disagrees with f on at most $\frac{\epsilon}{5}2^d$ elements of $H^d_{\epsilon/10}$. The number of $x \in \{0,1\}^d$ not in $H^d_{\epsilon/10}$ is at most $\frac{\epsilon}{10}2^d$, so overall f_{mon} disagrees with f on at most $\frac{3\epsilon}{10}2^d$ elements of $\{0,1\}^d$, in other words $\|f - f_{\text{mon}}\| \leq \frac{3\epsilon}{10}$. Via triangle inequality, we have $\|g - f_{\text{mon}}\| \leq \|g - f\| + \|f - f_{\text{mon}}\| \leq \frac{2\epsilon}{5} \leq \epsilon$, finishing the proof. □

Let us remark on the performance of our algorithm in the agnostic setting. Observation 3 on page 5 of [KKMS08] implies that the algorithm of [BT96] (i.e. Theorem 27), when run in the agnostic setting, will give a predictor with error at most $8 \cdot \text{opt} + \epsilon$, where opt is the error of best monotone predictor. Repeating the argument in the proof of Theorem 30 then tells us that in this setting our proper learning algorithm will also have prediction error $C \cdot \text{opt} + \epsilon$ for some absolute constant C. In particular, this means[9] in the agnostic learning model of Kearns, Schapire, and Sellie [KSS94b], our algorithm can handle a noise rate of $\Omega(\epsilon)$.

We now present how to obtain a better error guarantee in the agnostic setting at a cost of slightly worse dependence of run-time on ϵ:

Theorem 31 (Proper learning in agnostic setting)**.** *In the agnostic setting with examples distributed uniformly over $\{0,1\}^d$, there is a learning algorithm that outputs a circuit computing a **monotone function** $\widehat{g}$, such that if the best monotone predictor has error* opt*, then the error of the predictor $\widehat{g}$ is at most $3 \cdot \text{opt} + \epsilon$. The algorithm uses $d^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ samples and run-time. The failure probability of the algorithm is at most $1/2^d$.*

Proof. The proof, presented in Section 4.5.3, follows a pattern similar to the proof of Theorem 30. □

4.4 The LCA for poset sorting

In this section, we prove Theorem 29. First, we give a "global" algorithm for the poset sorting problem, which reads all the values of f and writes all the values of f_{mon}. The global algorithm is inefficient, but lends itself easily to a proof of correctness. We prove correctness for the global algorithm, then give our local implementation and show that it simulates the global algorithm.

4.4.1 A global view

We first present Algorithm 7, which sorts the labels of f in stages by swapping the labels of pairs of vertices that violate monotonicity. We will show that each stage reduces the maximum

[9] Let us elaborate. If we take the error parameter in our algorithm to be $\epsilon/2$, then we see that our algorithm will have prediction error $C \cdot \text{opt} + \epsilon/2$. Then, if noise rate opt is $\frac{\epsilon}{2C}$ or less, our predictor will be ϵ-competitive with the best monotone function, as required by the agnostic learning model of Kearns, Schapire, and Sellie [KSS94b].

distance between violated vertices by a factor of 2, which produces a monotone function after $\log h$ stages, where h is the height of the input graph.

Before we present the algorithm, we define the following objects that it constructs during its execution.

Definition 22 (Violation set)**.** *We define the set of violated pairs* $\mathrm{viol}_P(f)$ *as follows:*

$$\mathrm{viol}_P(f) := \{(v, w) \in P \times P : \ v \succ w, \ f(v) = 0 \text{ and } f(w) = 1\}.$$

Definition 23 (k-violation graph)**.** *For a poset* P *of height* h*, a function* $f : P \to \{0, 1\}$*, and some* $k \in [h]$*, we define the* k*-violation graph* B_k *as follows:*

- $V(B_k) = P$*, and*
- *For* $(v, w) \in \mathrm{viol}_P(f)$*,* $(v, w) \in E(B_k)$ *iff* $\mathrm{dist}(v, w) \geq k$*.*

Note that B_k *is bipartite and undirected.*

Algorithm 7: LCA for sorting labels in a poset: the global view

Given: Poset P of height h, function $f_0 : P \to \{0, 1\}$
Output: monotone function over P
Let $i \leftarrow 0$
for $0 \leq i \leq \lceil \log h \rceil + 1$ **do**
  Let $k \leftarrow \lceil h/2^{i+1} \rceil$
  Construct the k-violation graph B_k from P and f_i.
  Compute a maximal matching in B_k and let λ map matched vertices to each other, and unmatched vertices to themselves.
  For all $x \in P$, let $f_{i+1}(x) = f_i(\lambda(x))$
  $i \leftarrow i + 1$
end for
return f_i

4.4.2 Correctness of Algorithm 7

Recall that we are required to show that our algorithm outputs a monotone function that can be obtained from f by a sequence of monotonicity-violating label swaps. Since it is evident from the pseudocode that this algorithm only performs such swaps, it remains to show only that the output is monotone.

Our algorithm works by finding a maximal matching over the k-violation graph B_k, and swapping the matched labels. We first claim that performing this swap reduces the distance (length of the longest path) between violated labels by at least k.

Lemma 22 (Distance shortening lemma)**.** *Let P be any poset. Let f be a $\{0,1\}$-valued function over P and k be a positive integer. Let B_k be as defined in Definition 23. Suppose one picks some maximal matching M over B_k and obtains a new function f' as follows*

$$f'(x) = \begin{cases} f(x) & \text{if } x \text{ was not matched}, \\ f(y) & \text{if } x \text{ was matched to some } y. \end{cases}$$

Then, we have

$$\max_{(\boldsymbol{v},\boldsymbol{w})\in \mathrm{viol}_P(f')} \mathrm{dist}(v,w) \leq$$
$$\max\left(k-1, \left(\max_{(\boldsymbol{v},\boldsymbol{w})\in \mathrm{viol}_P(f)} \mathrm{dist}(v,w)\right) - k\right).$$

Proof. Let $\lambda : P \to P$ map x to (i) y if x was mapped to some y by M (ii) x itself otherwise. Note that $f'(x) = f(\lambda(x))$ and also λ is one-to-one.

Let x, y in P be such that $x \succ y$, $f(x) = 0$ and $f(y) = 1$. If $\mathrm{dist}(x,y) \geq k$, it cannot be the case that both $\lambda(x) = x$ and $\lambda(y) = y$, because then M would not be a maximal matching since we could also match x to y. Besides, note that if $\lambda(x) \neq x$ then $\lambda(x) \prec x$ and $\mathrm{dist}(x, \lambda(x)) \geq k$. Analogously, if $\lambda(y) \neq y$ then $\lambda(y) \succ y$ and $\mathrm{dist}(y, \lambda(y)) \geq k$. Additionally, for any $a, b, c \in P$ if $a \succ b \succ c$ then $\mathrm{dist}(a,c) \geq \mathrm{dist}(a,b) + \mathrm{dist}(b,c)$, as there exists a path from a to c that is the union of the longest paths from a to b and b to c. Taking these observations together, we see that only following eight cases are possible:

1. $\mathrm{dist}(x,y) \leq k-1$.
   (a) $\lambda(x) \succ \lambda(y)$ and $\mathrm{dist}(\lambda(x), \lambda(y)) \leq \mathrm{dist}(x,y) \leq k-1$.
   (b) It is not the case that $\lambda(x) \succ \lambda(y)$.
2. $\mathrm{dist}(x,y) \geq k$.
   (a) $\lambda(x) = x$ and $\mathrm{dist}(y, \lambda(y) \geq k$
      i. $\lambda(x) \succ \lambda(y)$ and $\mathrm{dist}(\lambda(x), \lambda(y)) \leq \mathrm{dist}(x,y) - k$.
      ii. It is not the case that $\lambda(x) \succ \lambda(y)$.
   (b) $\lambda(y) = y$ and $\mathrm{dist}(x, \lambda(x)) \geq k$
      i. $\lambda(x) \succ \lambda(y)$ and $\mathrm{dist}(\lambda(x), \lambda(y)) \leq \mathrm{dist}(x,y) - k$.
      ii. It is not the case that $\lambda(x) \succ \lambda(y)$.
   (c) $\mathrm{dist}(x, \lambda(x)) \geq k$ and $\mathrm{dist}(y, \lambda(y)) \geq k$
      i. $\lambda(x) \succ \lambda(y)$ and $\mathrm{dist}(\lambda(x), \lambda(y)) \leq \mathrm{dist}(x,y) - 2k$.

ii. It is not the case that $\lambda(x) \succ \lambda(y)$.

In the whole, if $\lambda(x) \succ \lambda(y)$ then $\mathrm{dist}(\lambda(x), \lambda(y)) \leq \max(k-1, \mathrm{dist}(x,y)-k)$.

Now let's consider what happens after the swap. Let v_0, w_0 in P maximize $\mathrm{dist}(v_0, w_0)$ subject to $v_0 \succ w_0$, $f'(v_0) = 0$ and $f'(w_0) = 1$. Let $x = \lambda^{-1}(v_0)$ and $y = \lambda^{-1}(w_0)$. Since $f'(v_0) = 0$, we have $x \succeq v_0$ and since $f'(w_0) = 1$, we have $y \preceq w_0$. Therefore, $x \succ y$. Also, $f(x) = f'(v_0) = 0$, $f(y) = f'(w_0) = 1$ and $\lambda(x) \succ \lambda(y)$. The conclusion of the previous paragraph tells us that $\mathrm{dist}(v_0, w_0) = \mathrm{dist}(\lambda(x), \lambda(y)) \leq \max(k-1, \mathrm{dist}(x,y)-k)$. Overall,

$$\begin{aligned}
&\max_{(v,w)\in \mathrm{viol}_P(f')} \mathrm{dist}(v,w) = \mathrm{dist}(v_0, w_0) \\
&\leq \max(k-1, \mathrm{dist}(x,y)-k) \\
&\leq \max\left(k-1, \left(\max_{(v,w)\in \mathrm{viol}_P(f)} \mathrm{dist}(v,w)\right) - k\right),
\end{aligned}$$

which finishes the proof. □

The following invariant, which will be useful for proving that the output is monotone, is a consequence of Lemma 22.

Corollary 10 (Distance shortening invariant)**.** *The following holds for all f_i, $0 \leq i \leq \lceil \log h \rceil$:*

$$\max_{(v,w)\in \mathrm{viol}_P(f)} \mathrm{dist}(v,w) \leq \left\lceil \frac{h}{2^i} \right\rceil$$

Proof. We proceed by induction on i. For $i = 0$, the distance must be at most h, because h is the height of P. Assume as an inductive hypothesis that the claim holds for some $i \leq \lceil \log h \rceil$.

B_k is a graph with the properties described in Definition 23: it has an edge joining each pair of vertices in P that violates monotonicity and has distance at least k, for $k = \lceil h/2^{i+1} \rceil$. By the inductive hypothesis, all such distances are between $\lceil h/2^{i+1} \rceil$ and $\lceil h/2^i \rceil$ inclusive. Then by Lemma 22 we guarantee that

$$\begin{aligned}
&\max_{(v,w)\in \mathrm{viol}_P(f)} \mathrm{dist}(v,w) \\
&\leq \max\left(\left\lceil \frac{h}{2^{i+1}} \right\rceil - 1,\ \left\lceil \frac{h}{2^i} \right\rceil - \left\lceil \frac{h}{2^{i+1}} \right\rceil\right) \\
&\leq \left\lceil \frac{h}{2^{i+1}} \right\rceil.
\end{aligned}$$

This completes the induction. □

With Corollary 10 in hand, we continue with the proof of correctness.

Lemma 23 (Correctness)**.** *For any Boolean function f_0, poset P, and any choice of maximal matchings over B_k in Algorithm 7, the output $f_{\lceil \log h \rceil + 1}$ is monotone over P.*

Proof. By Corollary 10, we have

$$\max_{(v,w)\in \mathrm{viol}_P(f_{\lceil \log h \rceil})} \mathrm{dist}(v,w) \leq \left\lceil \frac{h}{2^{\lceil \log h \rceil}} \right\rceil = 1$$

Then $f_{\lceil \log h \rceil}$ has the property that all pairs that violate monotonicity are immediate neighbors in P. By one more application of Lemma 22, we have

$$\max_{(v,w)\in \mathrm{viol}_P(f_{\lceil \log h \rceil+1})} \mathrm{dist}(v,w)$$
$$\leq \max\left(1-1, 1-\left\lceil \frac{h}{2^{\lceil \log h \rceil+1}} \right\rceil\right) = 0,$$

indicating that $f_{\lceil \log h \rceil+1}$ is monotone. □

4.4.3 Local implementation

In this section, we provide an LCA that gives membership query access to the output of Algorithm 7, and we analyze its complexity. To better explain how our LCA simulates Algorithm 7, we present it as a system of three LCAs, each parameterized by the iteration number i. Algorithm 8 makes queries to the i_{th} function f_i and an all-neighbors oracle for P, and answers queries to the i_{th} k-violation graph B_i. Algorithm 9 makes queries to B_i and answers queries to a maximal matching λ_i over it. Algorithm 10 makes queries to f_i and λ_i and answers queries to f_{i+1}, which swaps the matched labels.

Algorithm 8: An LCA for undirected all-neighbors queries $B_i(x, P, f_i, h, i)$

Given: Target vertex x, all-neighbors (immediate predecessor and successor) oracle for P, membership query oracle f_i, height h, iteration number i.
Initialize $k \leftarrow \lceil h/2^{i+1} \rceil$
if $f_i(x) = 1$ **then**
  $S \leftarrow$ the set of all successors of x
else
  $S \leftarrow$ the set of all predecessors of x
end if
Compute longest path $\mathrm{dist}(x, y)$ for each $y \in S$ by dynamic programming
Remove any y from S such that $\mathrm{dist}(x, y) < k$ or $f(y) = f(x)$

return S

Algorithm 9: An LCA for maximal matchings $\lambda_i(x, B_i, r, \delta)$

Given: Target vertex x, undirected all-neighbors query oracle B_i, random seed r, and confidence parameter δ.
Call the algorithm described in Corollary 7 with B_i as the graph, x as the target, and r as the random seed and Δ as the degree bound. If x is in the matching, return the vertex that it is matched to; otherwise return x.

Algorithm 10: An LCA for membership queries $f_{i+1}(x, \lambda_i, f_i)$

Given: Target vertex x, matching query oracle λ_i, membership query oracle f_i
return $f_i(\lambda_i(x))$

Analysis of our implementation

Throughout this section, for any algorithm A, the notation $T(A)$ denotes the running time of A.

Claim 6 (Behavior of B_i)**.** *For any $0 \le i \le \lceil \log h \rceil + 1$, B_i provides all-neighbors query access to the $\lceil h/2^{i+1} \rceil$-violation graph of P with respect to f_i. Furthermore, if each element of P has at most Δ successors or predecessors, then $T(B_i) \le O(\Delta \cdot T(f_i))$.*

Proof. If $f_i(x) = 1$, then all neighbors of x in the $\lceil h/2^{i+1} \rceil$-violation graph of P are successors of x. Finding all the successors takes $O(\Delta)$ time by depth-first search. Computing $\mathrm{dist}(x, y)$ for each successor y of x takes $O(\Delta)$ time by standard dynamic programming techniques for finding longest paths in a DAG. Comparing $f_i(x)$ to $f_i(y)$ takes $O(\Delta)$ queries to f_i, and therefore $O(\Delta \cdot T(f_i))$ time. The case of $f_i(x) = 0$ is symmetric. □

Claim 7 (Behavior of λ_i)**.** *For any $0 \le i \le \lceil \log h \rceil + 1$, and $\delta \in (0, 1]$, λ_i provides query access to a maximal matching over B_i with probability $1 - \delta/(\lceil \log h \rceil + 1)$, using a random seed of length $\mathrm{poly}(\Delta \log(N/\delta))$. Furthermore, $T(\lambda_i) \le \mathrm{poly}(\Delta \log(N/\delta)) \cdot T(B_i)$.*

Proof. Let $\delta' = \delta/(\lceil \log h \rceil + 1)$. By Corollary 7, λ_i fails with probability at most δ', where the query complexity and the length of the random seed are each $\mathrm{poly}(\Delta \log(N/\delta'))$. Since h is always at most Δ, this is still $\mathrm{poly}(\Delta \log(N/\delta))$. The claim follows from the fact that since the queries are made to B_i, each query takes time $T(B_i)$. □

We now proceed with our proof of Theorem 29, which we restate here for convenience.

Theorem 29. *Let P be a poset of N elements and height h, such that each element in P has at most Δ predecessors or successors. Suppose we are given query access to the graph of P, i.e. for any $x \in P$ we can obtain the immediate predecessors or successors of x. Also, suppose we are given query access to some Boolean function f over P. Then, there is an LCA that solves the poset-sorting problem for f over P: in other words, it provides query access to a monotone*

function $f_{mon} : P \to \{0,1\}$ that can be obtained from f by a sequence of swaps of monotonicity-violating label pairs. The LCA has query complexity and run-time of $\left(\Delta \log\left(\frac{N}{\delta}\right)\right)^{O(\log h)}$, and it uses a random bit-string of length $\mathrm{poly}(\Delta \log \frac{N}{\delta})$. The failure probability of the LCA is at most δ.

Proof. Assume that for each $i \leq \lceil \log h \rceil + 1$, the matching provided by λ_i is maximal for B_i. Under this condition, f_i, λ_i, and B_i implement LCAs for all the objects expected by Algorithm 7 (by Claim 6 and Claim 7). The correctness result of Lemma 23 implies that our implementation of $f_{\lceil \log h \rceil + 1}$ is an LCA for a function with the properties claimed in this theorem. We will bound the running time and query complexity by a recurrence relation.

As a base case, we will let $T(f_0)$ be 1. We have three recurrences: $T(B_i) \leq O(\Delta \cdot T(f_i))$, $T(\lambda_i) \leq \mathrm{poly}(\Delta \log(N/\delta)) \cdot T(B_i)$, and $T(f_{i+1}) \leq O(T(f_i) + T(\lambda_i))$. To simplify:

$$\begin{aligned} T(f_{i+1}) &\leq O(T(f_i) + T(\lambda_i)) \\ &\leq T(f_i) + \mathrm{poly}(\Delta \log(N/\delta)) \cdot T(B_i) \\ &\leq T(f_i) + \mathrm{poly}(\Delta \log(N/\delta)) \cdot \Delta T(f_i) \\ &\leq \mathrm{poly}(\Delta \log(N/\delta)) \cdot T(f_i) \end{aligned}$$

This recurrence resolves to

$$T(f_i) = (\Delta \log(N/\delta))^{O(i)}$$

for both running time and query complexity. Letting $i = \lceil \log h \rceil + 1$, we have a total running time and query complexity of $(\Delta \log(N/\delta))^{O(\log h)}$.

We initialize our algorithm with a random bit string of length $\mathrm{poly}(\Delta \log(N/\delta))$ as required by Claim 7. Each call to λ_i fails with probability $\delta/(\lceil \log h \rceil + 1)$; this gives a total failure probability of δ. Therefore, with probability at least $1 - \delta$ all the matchings are maximal and our analysis holds. □

4.5 Standard proofs

4.5.1 Proof of Corollary 7

For a graph G, one forms the so-called line graph of G, denoted as G', as follows: (i) the vertex set of G' is the edge set of G, (ii) two vertices are connected in G' if the corresponding edges in G share a vertex. One sees immediately that maximal matchings on G translate to maximal independent sets on G', and vice versa. Therefore, one can use the LCA of [GU19] (described here in Theorem 26) to get access to a maximal independent set in G', which will translate to a maximal matching on G.

An all-neighbor query to G' can be simulated via two all-neighbor queries to G. The graph G' has at most ΔN vertices and degree at most 2Δ. Overall, this means that the query complexity

and the run-time of the LCA are still $\mathrm{poly}(\Delta, \log(N/\delta))$, the length of the random bit-string is still $\mathrm{poly}(\Delta, \log(N/\delta))$ and the failure probability is still at most δ.

4.5.2 Proof of Corollary 9

First of all, without loss of generality we can assume $\delta = 2/3$, because error probability can be reduced via repetition.

Theorem 29 (via Remark 45) allows us to gain query access to f_{mon}, such that distance of f to f_{mon} is at most twice the distance of f to monotonicity. Then, obtaining the values of both these functions on i.i.d. uniformly random points of P, we estimate $\|f - f_{\text{mon}}\|$ up to error 0.005ϵ. Then, if the distance of f to monotonicity is at least ϵ, the distance $\|f - f_{\text{mon}}\|$ will be also at least ϵ, so the value of the estimate will be at least 0.995ϵ. On the other hand, if the distance of f to monotonicity is at most 0.49ϵ, then $\|f - f_{\text{mon}}\|$ will at most 0.98ϵ and the value of the estimate will be at most 0.985ϵ. Overall, checking if the value of the estimate is greater than 0.99ϵ we can see which of the two cases we are in.

For the estimation to succeed, we need to evaluate f and f_{mon} on $O\left(\frac{1}{\epsilon^2}\right)$ i.i.d. random elements of P[10]. Overall, this will take $\Delta^{O(\log h \log\log \Delta)} \left(\log\left(N\right)\right)^{O(\log h)} \frac{1}{\epsilon^2}$ queries and run-time.

4.5.3 Proof of Theorem 31

The proof follows a pattern similar to the proof of Theorem 30. The only modification to the algorithm in the proof of Theorem 30 is that in step 1 we obtain g by using the agnostic improper learner of Theorem 28 (instead of the learner of Theorem 27). The accuracy parameter there will still be $\frac{\epsilon}{10}$. With probability at least $1/2^d$ both algorithms we use succeed, which we will assume henceforth.

The run-time analysis remains the same, except the learner in Theorem 28 now takes $d^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon^2}\right)}$ samples and run-time (as opposed to $d^{O\left(\frac{\sqrt{d}}{\epsilon}\right)}$ samples and run-time for the learner in Theorem 27). Repeating the argument, the overall run-time and sample complexity is also $d^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon^2}\right)}$.

Finally, we argue correctness. The function f_{mon} computed by the circuit we output is again monotone by the same argument as in proof of Theorem 27. Theorem 28 tells us that function f has generalization error of at most $\mathrm{opt} + \frac{\epsilon}{10}$. Also, let f^* be a monotone function with the best available generalization error of opt. This implies that, f and f^* disagree on at most $\left(2\mathrm{opt} + \frac{\epsilon}{10}\right) 2^d$ elements[11].

Now, recall that the extra property of the LCA in Theorem 29 (noted in Remark 45) tells us that for any monotone function q over $H^d_{\epsilon/10}$, its distance to f_{mon} is at most its distance to f. Taking

[10] For the LCA of f_{mon} we can set the overall success probability parameter to be 0.1. A Chernoff bound and union bound argument then shows that overall success probability is at least $2/3$.

[11] Specifically, for a random example-label pair (x, y) (x distributed uniformly) we have $\Pr\left[f(x) \neq f^*(x)\right] = \Pr\left[f(x) = y,\ f^*(x) \neq y\right] + \Pr\left[f(x) \neq y,\ f^*(x) = y\right]$, which can be upper-bounded by $\Pr\left[f^*(x) \neq y\right] + \Pr\left[f(x) \neq y\right]$. Given the bounds we know for these probabilities, the bound on $\|f - f^*\|$ follows.

$q = f^*$, we get that f_{mon} and f^* can disagree only on at most $\left(2\text{opt} + \frac{\epsilon}{10}\right) 2^d$ elements of $H^d_{\epsilon/10}$. The number of $x \in \{0,1\}^d$ not in $H^d_{\epsilon/10}$ is at most $\frac{\epsilon}{10}2^d$, so overall f_{mon} disagrees with f^* on at most $\left(2\text{opt} + \frac{\epsilon}{5}\right) 2^d$ elements of $\{0,1\}^d$, in other words $\|f^* - f_{\text{mon}}\| \leq 2\text{opt} + \frac{\epsilon}{5}$. Via triangle inequality, the generalization error of f_{mon} is at most the sum of (i) generalization error of f^* and (ii) the distance between f_{mon} and f^*. This means that the generalization error of f_{mon} is at most $3\text{opt} + \frac{\epsilon}{5} \leq 3\text{opt} + \epsilon$, finishing the proof.

4.5.4 Refined local implementation.

Here we explain how an improved run-time can be achieved for posets with additional characteristics. We shall assume that each element of the poset P has at most d immediate predecessors and at most d immediate successors. Additionally, we assume that the poset P is graded. This is achieved again by implementing the algorithm in Section 4.4.1, but the run-time is somewhat faster than the one achieved in in Section 4.4.3. To be fully specific, Section 4.4.1 would give us a run-time of $d^{O(h \log h)}(\log(N/\delta))^{\log h}$ in this setting, which is here improved to $d^{O(h)}(\log(N/\delta))^{\log h}$.

Similarly to Section 4.4.3, we provide an LCA that gives membership query access to the output of Algorithm 7, and we analyze its complexity. This is again presented it as a system of three LCAs, each parameterized by the iteration number i. Algorithm 11 makes queries to the i_{th} function f_i and an all-neighbors oracle for P, and answers queries to the i_{th} k-violation graph B_i. Algorithm 12 makes queries to B_i and answers queries to a maximal matching λ_i over it. Algorithm 13 makes queries to f_i and λ_i and answers queries to f_{i+1}, which swaps the matched labels.

Algorithm 11: An LCA for undirected all-neighbors queries $B_i(x, P, f_i, h, i)$

Given: Target vertex x, immediate predecessor and successor oracle for P, membership query oracle f_i, height h, iteration number i.
Initialize $k \leftarrow \lceil h/2^{i+1} \rceil$
if $f_i(x) = 1$ **then**
  $S \leftarrow$ the set of all successors y of x such that $k \leq \text{dist}(x, y) \leq 2k$.
else
  $S \leftarrow$ the set of all successors y of x such that $k \leq \text{dist}(x, y) \leq 2k$.
end if
Note: because the poset is graded, the collections of successors and predecessors above can be found via breath-first search.
Remove any y from S such that $f(y) = f(x)$
return S

Algorithm 12: An LCA for maximal matchings $\lambda_i(x, B_i, r, \delta)$

Given: Target vertex x, undirected all-neighbors query oracle B_i, random seed r, and confidence parameter δ.
Call the algorithm described in Corollary 7 with B_i as the graph, x as the target, and r as the random seed and $d^{\lceil h/2^i \rceil}$ as the degree bound. If x is in the matching, return the vertex that it is matched to; otherwise return x.

Algorithm 13: An LCA for membership queries $f_{i+1}(x, \lambda_i, f_i)$

Given: Target vertex x, matching query oracle λ_i, membership query oracle f_i
return $f_i(\lambda_i(x))$

Analysis of our implementation

Throughout this section, for any algorithm A, the notation $T(A)$ denotes the running time of A. This is the same convention used in Section 4.4.3. Furthermore, the following two claims are analogous to Claim 6 and Claim 7 respectively.

Claim 8 (Behavior of B_i)**.** *For any $0 \leq i \leq \lceil \log h \rceil + 1$, suppose*

$$\max_{(v,w) \in \mathrm{viol}_P(f_i)} \mathrm{dist}(v, w) \leq \lceil h/2^i \rceil.$$

Then, B_i provides all-neighbors query access to the $\lceil h/2^{i+1} \rceil$-violation graph of P with respect to f_i. Furthermore, the degrees of all vertices in B_i are bounded by $d^{\lceil h/2^i \rceil}$ and $T(B_i) \leq O(d^{\lceil h/2^i \rceil} \cdot T(f_i))$.

Proof. If $f_i(x) = 1$, then all neighbors of x in the $\lceil h/2^{i+1} \rceil$-violation graph of P are successors of x. As, $\lceil h/2^i \rceil \leq 2\lceil h/2^{i+1} \rceil = 2k$, we see that initializing S to have only elements of distance at most $2k$ does not leave out any neighbors of x in the $\lceil h/2^{i+1} \rceil$-violation graph of P. Comparing with the definition of the $\lceil h/2^{i+1} \rceil$-violation graph of P, we see that the elements given by Algorithm 11 are precisely the neighbors of x in the $\lceil h/2^{i+1} \rceil$-violation graph of P.

Finally, the bound of $d^{\lceil h/2^i \rceil}$ on the degree of B_i follows, because each element in P has at most d immediate predecessors or successors. The bound on the run-time follows from the bound on degree of B_i. □

Claim 9 (Behavior of λ_i)**.** *For any $0 \leq i \leq \lceil \log h \rceil + 1$, and $\delta \in (0, 1]$, if degrees of all vertices in B_i are bounded by $d^{\lceil h/2^i \rceil}$, then λ_i provides query access to a maximal matching over B_i with probability $1 - \delta/(\lceil \log h \rceil + 1)$, using a random seed of length $\mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta))$. Furthermore, the run-time $T(\lambda_i)$ is bounded by $\mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \cdot T(B_i)$.*

Proof. Let $\delta' = \delta/(\lceil \log h \rceil + 1)$. By Corollary 7, λ_i fails with probability at most δ', where the query complexity and the length of the random seed are each $\mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta'))$. The claim follows from the fact that since the queries are made to B_i, each query takes time $T(B_i)$. □

We now proceed with our proof of Remark 5, that is to proving an overall run-time bound of $d^{O(h)}(\log(N/\delta))^{\log h}$.

Proof. We first argue, using an induction over i, that f_i, λ_i, and B_i implement LCAs for all the objects expected by Algorithm 7 (with overall probability of at least $1-\delta$) . The base case $i = 0$ is immediate. Suppose, this holds up to iteration i (i.e. condition on this event). Then, by Corollary 10 we have

$$\max_{(v,w)\in \mathrm{viol}_P(f_i)} \mathrm{dist}(v,w) \leq \lceil h/2^i \rceil,$$

so the premise of Claim 8 holds. Now, one of the conclusions of Claim 8 is that degrees of all vertices in B_i are bounded by $d^{\lceil h/2^i \rceil}$, which is the premise of Claim 9. Together, the conclusions of Claim 8 and Claim 9, imply that, with probability $1-\delta/(\lceil \log h \rceil + 1)$, B_i, λ_i and f_{i+1} implement the corresponding quantities expected by Algorithm 7. Via a union bound over all i we see that with overall probability of at least $1-\delta$ this indeed holds for all i.

We will bound the running time and query complexity by a recurrence relation. As a base case, we will let $T(f_0)$ be 1. We have three recurrences: $T(B_i) \leq O(d^{\lceil h/2^i \rceil} \cdot T(f_i))$, $T(\lambda_i) \leq \mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \cdot T(B_i)$, and $T(f_{i+1}) \leq O(T(f_i) + T(\lambda_i))$. To simplify:

$$\begin{aligned}
T(f_{i+1}) &\leq O(T(f_i) + T(\lambda_i)) \\
&\leq T(f_i) + \mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \cdot T(B_i) \\
&\leq T(f_i) + \mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \cdot d^{\lceil h/2^i \rceil} T(f_i) \\
&\leq \mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \cdot T(f_i)
\end{aligned}$$

Thus, we obtain

$$\begin{aligned}
T(f_i) &= \prod_{i=0}^{\lceil \log h \rceil} \mathrm{poly}(d^{\lceil h/2^i \rceil} \log(N/\delta)) \\
&= d^{O(h)}(\log(N/\delta))^{\log h}
\end{aligned}$$

for both running time and query complexity.

We note that our algorithm is initialized with a random bit string of length $d^{O(h)}\,\mathrm{polylog}(N/\delta)$, which is as required. □

Chapter 5

Agnostic proper learning of monotone functions: beyond the black-box correction barrier

.

5.1 Chapter Overview.

We note that Chapter 4 largely concerns itself with the problem of *realizable learning* of monotone functions, i.e. learning a function f that is itself promised to be monotone. In contrast, the focus of this chapter is the harder setting when the function f we access is *arbitrary* and we want to obtain a description of a monotone function g_{mon} that predicts f best among monotone functions (up to an additive slack of ϵ).

Specifically, in this chapter we consider two fundamental problems in this line of work: *approximating the distance* of unknown functions to monotone, and *agnostic proper learning* of monotone functions. For each of these problems we are given independent uniform samples $\{\boldsymbol{x}_i\}$ labeled by an arbitrary function $f : \{\pm 1\} \to \{\pm 1\}$ and we are required to perform the following tasks:

1. **Estimating distance to monotonicity** is the task of estimating up to some additive error ϵ the distance $\mathrm{dist}(f, f_{\text{mon}})$ from f to the monotone function f_{mon} that is closest to f.

2. **Agnostic proper learning of monotone functions** is the task of obtaining a description of a monotone function g_{mon}, whose distance $\mathrm{dist}(f, g_{\text{mon}})$ approximates $\mathrm{dist}(f, f_{\text{mon}})$ up to additive error ϵ.

Prior to this chapter, it was known that information-theoretically these tasks can be solved using only $2^{\tilde{O}\left(\sqrt{d}/\epsilon\right)}$ samples. However, all known algorithms had a run-time of $2^{\Omega(d)}$, thus dramatically exceeding the known sample complexity of $2^{\tilde{O}\left(\sqrt{d}/\epsilon\right)}$. In this chapter, we close this gap in our

knowledge and give algorithms for the two tasks above that not only use $2^{\tilde{O}(\sqrt{d}/\epsilon)}$ samples, but also run in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. This nearly matches the $2^{\tilde{\Omega}(\sqrt{d})}$ lower bound of [BCO+15].

5.1.1 Previous work

The results given in Chapter 4, give mixed additive-multiplicative approximation guarantees in the settings we study here. Specifically, Chapter 4 gives algorithms that also run in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$ and achieve the following:

1. Obtain a $(3, \epsilon)$-approximation of $\mathrm{dist}(f, f_{\text{mon}})$. In other words, the estimate is in the interval between $\mathrm{dist}(f, f_{\text{mon}})$ and $3 \cdot \mathrm{dist}(f, f_{\text{mon}}) + \epsilon$. (We also note that Chapter 4 additionally presents an algorithm that gives a distance estimate in $[\mathrm{dist}(f, f_{\text{mon}}), 2 \cdot \mathrm{dist}(f, f_{\text{mon}}) + \epsilon]$ but also requires query access to function f).

2. Obtain a succinct description of a monotone function g_{mon}, whose distance $\mathrm{dist}(f, g_{\text{mon}})$ is a $(3, \epsilon)$-approximation to $\mathrm{dist}(f, f_{\text{mon}})$. In other words, it is in the interval between $\mathrm{dist}(f, f_{\text{mon}})$ and $3 \cdot \mathrm{dist}(f, f_{\text{mon}}) + \epsilon$. As it is noted in Chapter 4, this yields a fully agnostic learning algorithm only if $\mathrm{dist}(f, f_{\text{mon}}) \leq O(\epsilon)$.

Overall, Table 5.1 summarizes how this chapter compares with Chapter 4 and other prior work.

Work	Guarantee for distance estimate and error for proper agnostic learning	Sample complexity	Run-time
[BT96, KKMS08] with refinement from [FKV17]	$[\mathrm{dist}(f, f_{\text{mon}}), \mathrm{dist}(f, f_{\text{mon}}) + \epsilon]$	$2^{\tilde{O}(\sqrt{d}/\epsilon)}$	$2^{\Omega(d)}$
Chapter 4	$[\mathrm{dist}(f, f_{\text{mon}}), 3 \cdot \mathrm{dist}(f, f_{\text{mon}}) + \epsilon]$	$2^{\tilde{O}(\sqrt{d}/\epsilon)}$	$2^{\tilde{O}(\sqrt{d}/\epsilon)}$
This chapter	$[\mathrm{dist}(f, f_{\text{mon}}), \mathrm{dist}(f, f_{\text{mon}}) + \epsilon]$	$2^{\tilde{O}(\sqrt{d}/\epsilon)}$	$2^{\tilde{O}(\sqrt{d}/\epsilon)}$

Table 5.1: Comparison of our results to previously known algorithms.

5.1.2 Main results

The following are our main results: learning and distance approximation of Boolean functions, and local correction of real-valued functions.

Theorem 32. *[Agnostic proper learning of monotone functions[1]] There is an algorithm that runs in time $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ and, given uniform sample access to an unknown function $f : \{\pm 1\}^d \to \{\pm 1\}$, with*

[1] See Section 5.6.3 for an extension to functions with randomized labels.

probability at least $1 - \frac{1}{2^d}$*, outputs a succinct representation of a monotone function* $g : \{\pm 1\}^d \to \{\pm 1\}$ *that is opt+*$O(\epsilon)$*-close to* f*, where opt is the distance from* f *to the closest monotone function (i.e. the fraction of elements of* $\{\pm 1\}^d$ *on which* f *and its closest monotone function disagree).*

The corollary below follows immediately by the standard method of [PRR04] that runs the learning algorithm in Theorem 32 and estimates the distance between g and f.

Corollary 11 (Additive distance-to-monotonicity approximation)**.** *There is an algorithm with running time and sample complexity* $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ *that outputs some estimate* est *of the distance from* f *to the closest monotone function* f_{mon}*. With probability at least* $1 - 2^{-d+1}$*, this estimate satisfies the guarantee*

$$\mathrm{dist}(f, f_{mon}) \leq est \leq \mathrm{dist}(f, f_{mon}) + O(\epsilon).$$

Our main result, Theorem 32, builds on an algorithm that is also of independent interest. It is a local computation algorithm for solving the "poset sorting problem" as described in Chapter 4 for real-valued functions (note that Chapter 4 only handled Boolean-valued functions). In other words, the algorithm gives local access to a monotone approximation of a real-valued function that is close to the optimal monotone approximation in ℓ_1 distance. (See Section 5.1.3 for background on local computation algorithms.)

Theorem 33. *[Local monotonicity correction of real-valued functions] Let* P *be a poset with* N *elements, such that every element has at most* Δ *predecessors or successors and the longest directed path has length* h*. Let* $f : P \to [-1, 1]$ *be* α*-close to monotone in* ℓ_1 *distance. There is an LCA that makes queries to* f *and outputs queries to* $g : P \to [-1, 1]$*, such that* g *is monotone and* $||f - g||_1 \leq 2\alpha + 3\epsilon$*. The LCA makes* $(\Delta \log N)^{O(\log h \log(1/\epsilon))}$ *queries, uses a random seed of length* $\mathrm{poly}(\Delta \log N)$*, and succeeds with probability* $1 - N^{-10}$*.*

5.1.3 Our techniques: beyond the black-box correction barrier.

The algorithms in Chapter 4 follow the following pattern (which we also summarize in Figure 5-1):

1. Use [BT96, KKMS08, FKV17] to obtain a succinct description of a (possibly non-monotone) function f_{improper} whose distance $\mathrm{dist}(f, f_{\text{improper}})$ is at most $\mathrm{dist}(f, f_{\text{mon}}) + \epsilon$. The issue now is that f_{improper} is not necessarily monotone, and therefore the distance $\mathrm{dist}(f, f_{\text{improper}})$ might dramatically underestimate the true distance to monotonicity $\mathrm{dist}(f, f_{\text{mon}})$.

2. Design and use a monotonicity corrector, in order to transform the succinct description of f_{improper} into a succinct description of some **monotone** function g_{mon} that is close to f_{improper}. Formally, Chapter 4 develops a corrector that guarantees that the distance $\mathrm{dist}(f_{\text{improper}}, g_{\text{mon}})$ satisfies

$$\mathrm{dist}(f_{\text{improper}}, g_{\text{mon}}) \leq c \min_{\text{monotone f'}} \mathrm{dist}(f_{\text{improper}}, f') + \epsilon, \tag{5.1}$$

   where the constant c is 2. They achieve this by a novel use of **Local Computation Algorithms** (LCAs) on graphs.

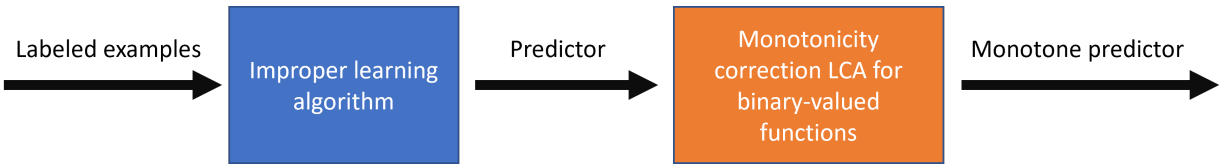

Figure 5-1: Control-flow diagram of the semiagnostic algorithm of Chapter 4

This way, Chapter 4 obtains a succinct polytime-evaluable description of a monotone function g_{mon} for which[2] $\text{dist}(f, g_{\text{mon}}) \leq 3 \cdot \text{dist}(f, f_{\text{mon}}) + \epsilon$.

However, one can see that even if the correction constant c in Equation (5.1) were equal to 1 (which is the best it can be) this approach could only yield a guarantee no better than $\text{dist}(f, g_{\text{mon}}) \leq (2 - o(1)) \cdot \text{dist}(f, f_{\text{mon}}) + \epsilon$. In particular, we claim that for a randomized function f, no black-box correction approach can give a classifier whose error is $(2 - \Omega(1)) \cdot \text{dist}(f, f_{\text{mon}})$. Formally, no algorithm that receives the best non-monotone predictor g for a randomized function f can produce, only on the basis of the predictor g and no additional information about f, a predictor $\widehat{f}$ satisfying $\text{dist}(f, \widehat{f}) \leq (2 - \Omega(1)) \cdot \text{dist}(f, f_{\text{mon}})$. This follows from the following argument. For an odd value of d, consider the following two random-valued functions f_1 and f_2 over $\{\pm 1\}^d$ defined as follows[3]:

- $f_1(x) = -1$ always for x satisfying $\sum_i x_i > 0$, and for x satisfying $\sum_i x_i < 0$ we have $f_1(x) = +1$ with probability $0.5 + o(1)$ and $f_1(x) = -1$ with probability $0.5 - o(1)$.
- $f_2(x) = +1$ always for x satisfying $\sum_i x_i < 0$, and for x satisfying $\sum_i x_i > 0$ we have $f_2(x) = -1$ with probability $0.5 + o(1)$ and $f_2(x) = -1$ with probability $0.5 - o(1)$.

Denoting $f_{1,\text{mon}}$ and $f_{2,\text{mon}}$ the best monotone predictors for f_1 and f_2 respectively, we see that $f_{1,\text{mon}}$ is the function that takes the value -1 on all elements of $\{\pm 1\}^d$ and $f_{2,\text{mon}}$ is the function that takes the value $+1$ on all elements of $\{\pm 1\}^d$. Overall we have $\text{dist}(f_1, f_{1,\text{mon}}) = 0.25 + o(1)$ and $\text{dist}(f_2, f_{2,\text{mon}}) = 0.25 + o(1)$. Let g be the function that maps to -1 values of x with $\sum_i x_i > 0$ and to $+1$ values of x with $\sum_i x_i < 0$. We see that, for both f_1 and f_2, the function g is the best among all general predictors (i.e. predictors that are not necessarily monotone). However, we claim that no algorithm can transform g into a monotone function $\widehat{f}$ that both satisfies $\text{dist}(f_1, \widehat{f}) \leq (2 - \Omega(1)) \cdot \text{dist}(f_1, f_{1,\text{mon}}) = 0.5 - \Omega(1)$ and $\text{dist}(f_2, \widehat{f}) \leq (2 - \Omega(1)) \cdot \text{dist}(f_2, f_{2,\text{mon}}) = 0.5 - \Omega(1)$, because we claim that no such monotone function $\widehat{f}$ exists. Indeed, let $\alpha \in [0, 0.5]$ denote the probability $\Pr_{x \sim \{\pm 1\}^d}[\sum_i x_i > 0 \text{ and } \widehat{f} = +1]$. From monotonicity of $\widehat{f}$, we see that $\Pr_{x \sim \{\pm 1\}^d}[\sum_i x_i < 0 \text{ and } \widehat{f} = -1] \geq 0.5 - \alpha$. Overall, we see that $\text{dist}(f_1, \widehat{f}) \geq \alpha + 0.25 - o(1)$ and that $\text{dist}(f_1, \widehat{f}) \geq 0.75 - \alpha - o(1)$. Overall, for every α in $[0, 0.5]$, at least one of these quantities is at least $0.5 - o(1)$.

[2]Strictly speaking, the properties of the corrector described so far yield only a guarantee of $\text{dist}(f, g_{\text{mon}}) \leq 4 \cdot \text{dist}(f, f_{\text{mon}}) + \epsilon$. To improve the multiplicative error constant from 4 to 3 the work in Chapter 4 uses an additional property of the corrector.

[3]Note that the functions f_1 and f_2 are randomized functions that can map the same value of x to $+1$ or to -1 with some probability depending on x. See Section 5.6.3 for more information on randomized functions.

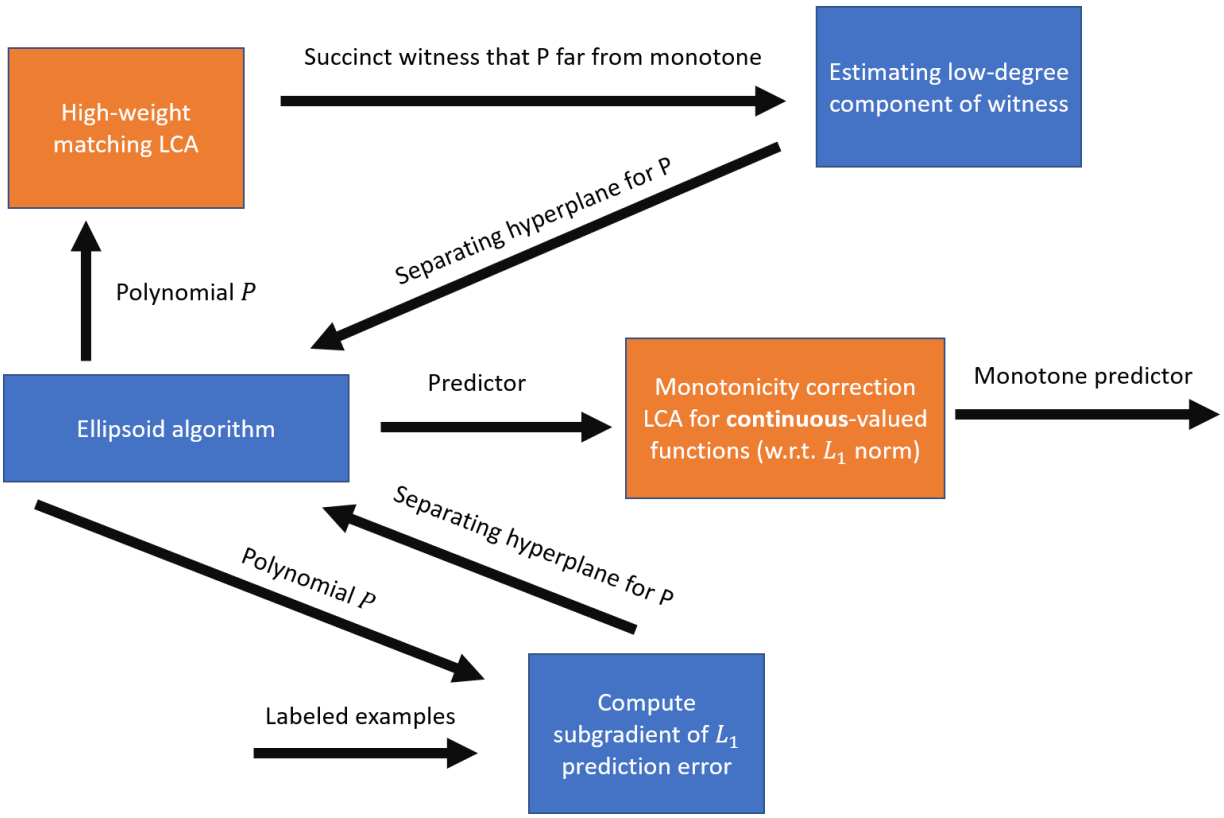

Figure 5-2: Control-flow diagram of the fully agnostic learning algorithm presented in this chapter (the final rounding step is omitted).

Description of our approach

In this chapter, we overcome the black-box correction barrier by using a different approach, summarized in Figure 5-2. As before, there is an improper learning phase and a correction phase; however in both phases we work with real-valued functions. We have essentially three steps:

1. Find a real-valued polynomial P that is ϵ-close to some monotone function, $(\mathrm{opt}+\epsilon)$-close[4] to the unknown function f in ℓ_1 distance, and bounded in $[-1,1]$.
2. Obtain a succinct description of a real-valued function $P_{\text{CORRECTED}}$ that is monotone, and $O(\epsilon)$-close to P in ℓ_1 distance.
3. Round the real-valued function $P_{\text{CORRECTED}}$ to be $\{\pm 1\}$-valued, while preserving monotonicity and closeness to f.

In contrast to the approach of Chapter 4, the improper learning phase is constrained to produce a good predictor that is ϵ-close to some monotone function, regardless of how far f may be from

[4]Since opt is unknown, we instead guess values of opt in increments of ϵ.

monotone. Existing improper learning algorithms are far from satisfying this new requirement. We design a new improper learner by combining the polynomial-approximation based techniques of [BT96, KKMS08, FKV17] with graph LCAs and the *ellipsoid method* for convex optimization.

The improper learning task is a convex feasibility problem; the set of polynomials satisfying the constraints we give in step (1) is a convex subset of the initial convex set of low-degree real polynomials. The ellipsoid method requires a *separation oracle*, i.e. some way to efficiently generate a hyperplane separating a given infeasible polynomial from the feasible region. Such hyperplanes are themselves low-degree real polynomials, which have high inner product with the infeasible polynomial and low inner product with every point in the feasible region. The separator for the set of polynomials that are $(\alpha + \epsilon)$-close to f is, as shown in Figure 5-2, just the gradient of the prediction error; the more interesting case is the separator for the set of polynomials that are ϵ-close to monotone.

With an argument inspired by the characterization of Lipschitz functions given in [BRY14a], we observe that if a real-valued polynomial P is far from monotone, this can be witnessed by a large matching on the pairs of elements on which P violates monotonicity. Given any description of the matching, we show how to extract a separating hyperplane for P by evaluating the matching on a set of sample points. Therefore, the challenge is to find a description of a sufficiently large matching that can also be evaluated quickly. We elaborate on this in the next section.

Step (2) requires another technical contribution, which is an extension of the poset-sorting LCA of Chapter 4 to real-valued functions. This extension is crucial for us to achieve the overall agnostic learning guarantee, because in the improper learning phase we obtain a real-valued function that is only close to monotone in ℓ_1 distance.[5] For step (3) we use the rounding procedure of [KKMS08] that rounds real-valued functions to $\{\pm 1\}$-valued functions, and we show that this procedure also preserves monotonicity.

LCAs and succinct representations of large objects

In this chapter we employ heavily the concept of a *succinct representation*. The succinct representations we deal with will have size and evaluation time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. To be fully specific, we consider succinct representations of two types of objects:

- A succinct representation of a function $f : \{\pm 1\}^d \to \mathbb{R}$ is an algorithm that, given $x \in \{\pm 1\}^d$, computes $f(x)$ in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$.
- A succinct representation of a (possibly weighted) graph G with the vertex set $\{\pm 1\}^d$ is an algorithm that, given $v \in \{-1, 1\}^d$, outputs all its neighbors and the weights of corresponding edges in time $2^{\tilde{O}(\sqrt{d}/\epsilon)}$.

A polynomial of degree $O(\sqrt{d})$ is an example of a succinct representation, but another type of representation that makes frequent appearances in this chapter is a *local computation algorithm*, or

[5] One can construct functions that are arbitrarily close to monotone in ℓ_1 norm but a constant fraction of their values needs to be changed for them to become monotone. Because of this, the corrector of Chapter 4 was not fit for our correction stage.

LCA [ARVX12b, RTVX11b]. An LCA efficiently computes a function over a large domain. For example, an LCA for an independent set takes as input some vertex v, makes some lookups to the adjacency list of the graph, then outputs "yes" or "no" so that the set of vertices for which the LCA would output "yes" form an independent set. Typically, its running time and query complexity are each sublinear in the domain size. We require that all LCAs used in this chapter have outputs consistent with one global object, regardless of the order of user queries, and without remembering any history from previous queries. This property allows us to use the LCA, in conjunction with any succinct representation of the graph, as a succinct representation of the object it computes. We formalize this relationship in Section 5.2.4.

5.1.4 Other related work

The local correction of monotonicity was studied in [ACSL08, SS10b, BGJ$^+$10, AJMR14] and in Chapter 4 of this thesis (see Chapter 4 for an overview of previously available algorithms for monotonicity correction and lower bounds).

The work of [CGG$^+$17] gives an improper learning algorithm for a function class that is larger than monotone functions. Additionally, we note that testing of monotone functions has also been studied over hypergrids [CS13b, BRY14a, BCS18, BCS20].

In addition to [Gha22], there have been many exciting recent works on local computation algorithms (LCAs). Some examples include [RTVX11b], [ARVX12b], [LRY17], [GHL$^+$15], [RV16], [EMR14], [Gha15], [CFG$^+$19], [ELMR21] , [PRVY19], [GU19],[LRR20], [AL21], [BGR21] and [GR21].

5.2 Preliminaries

5.2.1 Posets and $\{-1, 1\}^d$

Let P be a partially-ordered set. We use $\preceq$ to denote the ordering relation on P. We say $x \prec y$ ("x is a predecessor of y") if $x \preceq y$ and $x \neq y$, and use the analogous symbols $\succeq$ and $\succ$ for successorship. If $x \prec y$ and there is no z in P for which $x \prec z \prec y$, then x is an *immediate predecessor* of y and y is an *immediate successor* of x. We refer to the poset P and its Hasse diagram (DAG) interchangeably. The transitive closure $TC(P)$ is the graph on the elements of P that has an edge from each vertex to each of its successors. A *succinct representation* of P with size s is any computational procedure whose description is stored in s bits of memory that takes a vertex as input, outputs the sets of immediate predecessors and immediate successors, and runs in time $O(s)$ in the worst case over vertices.

Specific posets of interest in this chapter are the Boolean cube and the weight-truncated cube. We give a definition and a size-$O(d/\epsilon)$ representation computing the truncated cube.[6]

[6]See Algorithm 14 for the computational procedure that provides access to immediate successors and predecessor of a given element. Note that only size $O(d/\epsilon)$ is necessary because one can, for example, store a circuit that

Definition 24. *The* d-dimensional Boolean hypercube *is the set* $\{-1,1\}^d$*. For* $x, y \in \{-1,1\}^d$*, we say* $x \preceq y$ *if for all* $i \in \{1, \cdots, d\}$ *one has* $x_i \leq y_i$*. It is immediate that* $\{-1,1\}^d$ *is a poset with* 2^d *elements.*

We also define the truncated hypercube

$$H^d_\epsilon := \left\{ x \in \{-1,1\}^d : \left| \sum_i x_i \right| \leq \sqrt{2n \log \frac{2}{\epsilon}} \right\},$$

Via Hoeffding's bound, we have that the fraction of elements in $\{0,1\}^d$ *that are not also in* H^ϵ_n *is at most* $2\exp\left(-\frac{2t^2}{4n}\right) = \epsilon$*.*

Algorithm 14: LCA: TRUNCATEDCUBE(x, ϵ)

Given: Input $x \in \{-1,1\}^d$, truncation parameter ϵ

return $\{y \mid y$ differs from x in one bit and $|\sum_j y_j| \leq \sqrt{2n \log \frac{2}{\varepsilon}}\}$

Fourier analysis over $\{\pm 1\}^n$**.**

Let $[n]$ denote the set $\{1, 2, \cdots, n\}$. We define for every $S \subseteq [n]$ the function $\chi_S : \{\pm 1\}^n \to \mathbb{R}$ as $\chi_S(\boldsymbol{x}) := \prod_{i \in S} x_i$. We define the inner product between two functions $g_1, g_2 : \{\pm 1\}^n \to \mathbb{R}$ as follows: $\langle g_1, g_2 \rangle := \mathbb{E}_{\boldsymbol{x} \sim \{\pm 1\}^d}[g_1(\boldsymbol{x}) g_2(\boldsymbol{x})]$. It is known that $\langle \chi_{S_1}, \chi_{S_2} \rangle = \mathbb{1}_{S_1 = S_2}$. For a function $g : \{\pm 1\}^d \to \mathbb{R}$ we denote $\widehat{g}(S) := \langle g, \chi_S \rangle$. It is known that

$$g(\boldsymbol{x}) = \sum_{S \subseteq [n]} \widehat{g}(S) \chi_S(\boldsymbol{x}) \qquad\qquad \langle g_1, g_2 \rangle = \sum_{S \subseteq [n]} \widehat{g_1}(S) \widehat{g_2}(S).$$

5.2.2 Monotone functions

Part of our algorithm concerns monotonicity of functions over general posets. For a function $f : P \to \mathbb{R}$, we say that a pair of elements $x, y \in P$ forms a *violated pair* if we have $x \preceq y$ but $f(x) > f(y)$, and we define the *violation score* $\mathrm{vs}(x, y) := f(x) - f(y)$. The *violation graph* $\mathrm{viol}(f)$ is the subgraph of $TC(P)$ induced by violated pairs in f. The weight of an edge is the difference $f(x) - f(y)$.

The ℓ_1 distance of f to monotonicity $\mathrm{dist}(f, \mathrm{mono})$ is the ℓ_1 distance of f to the closest real-valued monotone function.

implements Algorithm 14.

Definition 25 (Distance to monotonicity)**.** *The ℓ_1 distance of $f : P \to \mathbb{R}$ to monotonicity is its distance to the closest real-valued monotone function.*

$$\mathrm{dist}_1(f, \mathrm{mono}) := \min_{\text{monotone } g: P \to \mathbb{R}} \left[\frac{1}{|P|} \sum_{x \in P} |f(x) - g(x)| \right]$$

The Hamming distance to monotonicity of $f : P \to \{-1, 1\}$ is defined analogously.

$$\mathrm{dist}_0(f, \mathrm{mono}) := \min_{\substack{\text{monotone } g: \\ P \to \{-1,1\}}} \left[\frac{1}{|P|} \sum_{x \in P} \mathbb{1}[f(x) \neq g(x)] \right]$$

We will need a bound on how well monotone functions can be approximated by low-degree polynomials. The following fact follows[7] from [BT96, KKMS08] and a refinement by [FKV17].

Fact 7. *For every monotone $f : \{-1, 1\}^d \to \{-1, 1\}$ and $\epsilon > 0$, there exists a multilinear polynomial p of degree $\lceil \frac{4 \cdot \sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \rceil$ such that*

$$||f - p||_1 \leq \epsilon.$$

5.2.3 Convex optimization

The following notion is standard in convex optimization.

Definition 26. *A **separation oracle** for a convex set $\mathcal{C}_{convex}$ is an oracle that given a point $\boldsymbol{x}$ does one of the following things:*

- *If $x \in \mathcal{C}_{convex}$, then the oracle outputs "Yes".*
- *If $x \notin \mathcal{C}_{convex}$, then the oracle outputs $(do, Q_{separation})$, where $Q_{separation} \in \mathbb{R}^d$ represents a direction along which x is separated from $\mathcal{C}_{convex}$. Formally, $\langle Q_{separation}, x \rangle > \langle Q_{separation}, x' \rangle$ for any x' in $\mathcal{C}_{convex}$.*

We will need the following well-known fact from convex optimization:

Fact 8. *[Kha80] There is an algorithm* EllipsoidAlgorithm *that takes as inputs positive real values r and R, and access to a separation oracle for some convex set $\mathcal{C}_{convex} \subset \{x \in \mathbb{R}^d : \quad \|x\| \leq R\}$. The algorithm runs in time $\mathrm{poly}\left(d, \log \frac{R}{r}\right)$ and either outputs an element in $\mathcal{C}_{convex}$ or outputs FAIL. Furthermore, if $\mathcal{C}_{convex}$ contains a ball of radius r, the algorithm is guaranteed to succeed.*

Also see [LSW15] for an overview of algorithms building on [Kha80].

[7] see Chapter 4 for more explanation on how these references yield the fact below.

5.2.4 LCAs and succinct representations

We use the following LCAs in this chapter[8] :

Theorem 34 (LCA for maximal matching [Gha22])**.** *There is an algorithm* GHAFFARIMATCHING *that takes adjacency lists access to a graph G, with N vertices and largest degree at most Δ, a random string $r \in \{0,1\}^{\mathrm{poly}(\Delta,\log(N/\delta))}$, parameter $\delta \in (0,1)$ and a vertex $v \in G$. The algorithm outputs the identity of a vertex $u : (u,v) \in E(G)$ or $\perp$. The algorithm runs in time $\mathrm{poly}(\Delta, \log(N/\delta))$ and with probability at least $1-\delta$ over the choice of r the condition of* ***global consistency holds*** *i.e. the set of edges $\{(u,v) \in G :$ GHAFFARIMATCHING$(G,r,\delta,u) = v\}$ is a maximal matching in the graph G.*

Theorem 35 (LCA for monotonicity correction of Boolean-valued functions from Chapter 4)**.** *There is an algorithm* BOOLEANCORRECTOR *that takes access to a function $f : P \to \{-1,1\}$ and adjacency lists access to a poset P with N vertices, such that each element has at most Δ predecessors and successors and the longest directed path has length h, a random string $r \in \{0,1\}^{\mathrm{poly}(\Delta,\log(N/\delta))}$, a parameter $\delta \in (0,1)$ and an element x in P. The algorithm outputs a value in $\{-1,1\}$. The algorithm runs in time $\Delta^{O(\log h)} \cdot \mathrm{polylog}(N/\delta)$ and with probability at least $1-\delta$ over the choice of r the condition of* ***global consistency holds*** *i.e. the function $g : P \to \{-1,1\}$ defined as $g(x) :=$ BOOLEANCORRECTOR(P,r,δ,x) is monotone and is such that $\Pr_{x\sim P}[g(x) \neq f(x)] \leq 2 \cdot \mathrm{dist}(f, \mathrm{mono})$.*

An important idea in Chapter 4 is that LCAs (i.e. algorithms that achieve global consistency) can be used to operate on succinct representations of combinatorial objects. To explain further, we need the following definition:

Definition 27 (Succinct representation)**.** *A succinct representation of a function f of size s is a description of f that is stored in s bits of memory and can be evaluated on an input in $O(s)$ time.*

For example, circuits of size s and polynomials of degree $\log s$ are examples of succinct representations of size s and $d^{\log s}$ respectively. The following fact follows immediately from the definition:

Fact 9 (Composition of representations)**.** *If a function f has a description that uses t bits of memory and evaluates in time $O(t)$ given q oracle queries to a function g, and g has a succinct representation of size s, then there is a succinct representation of f of size $O(t+sq)$.*

Now, for example, combining[9] Fact 9 and Theorem 34 we see immediately that for a graph G, with N vertices and largest degree at most Δ, using the algorithm in Theorem 34 we can

[8]To be fully precise, [Gha22] gives an LCA for the task of maximal independent set. The reduction to maximal matching is standard, see e.g. Chapter 4.

[9]A note on the description sizes of LCAs: because LCAs are uniform (i.e. Turing-machine) algorithms, they can be simulated with a uniform circuit family. For each input size, the size of the corresponding circuit is polynomial in the running time of the LCA for that input size.

transform a size-s representation[10] of a function computing all-neighbor access to G into a size-$\left(\Delta^{O(\log h)} \cdot \mathrm{polylog}(N/\delta) \cdot s\right)$ representation[11] of a function that determines membership in some maximal matching over G. Note that this transformation itself runs in time $\Delta^{O(\log h)} \cdot \mathrm{polylog}(N/\delta) \cdot s$. Analogously, in an exact same fashion it is possible to combine Fact 9 and Theorem 35.

5.3 Our algorithms

In this section we give descriptions of the agnostic learning algorithm and its major components (we will analyze the algorithms in the subsequent sections). The algorithm MONOTONELEARNER makes calls to
ELLIPSOIDALGORITHM, where the optimization domain is the $\leq d^{\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}$-dimensional space of degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomials over $\mathbb{R}^d$, and constraints given by $\textsc{Oracle}_{\alpha,d,\epsilon}$. It also makes calls to HYPERCUBECORRECTOR, which is given in Corollary 13.

The subroutine ORACLE takes as input a polynomial and provides the separating hyperplane required by ELLIPSOIDALGORITHM. It makes calls to HYPERCUBEMATCHING (see Lemma 31), which provides a high-weight matching over the pairs of labels that violate monotonicity.

The algorithm MATCHVIOLATIONS finds a high-weight matching on the violation graph of a poset. It is the main component of HYPERCUBEMATCHING, which is just a wrapper that calls MATCHVIOLATIONS on the truncated cube. FILTEREDGES removes vertices that are either incident to M or have weight below the threshold t, and GHAFFARIMATCHING is the maximal matching algorithm of Theorem 34. More implementation details and analysis are given in Section 5.5.

The following is the core of HYPERCUBECORRECTOR, given as a "global overview" for convenience. Analysis and local implementation are given in Section 5.4. The algorithm corrects monotonicity of a k-valued function over a poset. HYPERCUBECORRECTOR is a wrapper that discretizes a real-valued function and then calls this corrector with the truncated hypercube as the poset.

5.4 Analysis of the local corrector

In this section, we prove Theorem 33 by analyzing our algorithm for correcting a real-valued function over a poset in a way that preserves the ℓ_1 distance to monotonicity within a factor of 2. This extends the monotonicity corrector of Chapter 4 to handle functions with non-Boolean ranges.

Lemma 24 (ℓ_1 correction of k-valued functions)**.** *Let P be a poset and $f : P \to [k]$ be α-close to monotone in ℓ_1 distance. There is an LCA that makes queries to f and outputs queries to $g : P \to [k]$, such that g is monotone and $||f - g||_1 \leq 2\alpha$. The LCA makes $(\Delta \log N)^{O(\log h \log k)}$*

[10] For simplicity, in the rest of the chapter we will refer to such functions as a "succinct representation of G."

[11] For simplicity, in the rest of this chapter we will refer to such functions simply as "representation of a maximal matching."

Algorithm 15: Algorithm MONOTONELEARNER (d, ϵ, T)

Given: Integer d, $\epsilon \in (0,1)$, and uniform sample access to unknown $f : \{\pm 1\}^d \to \{\pm 1\}$.
Output: Circuit $\mathcal{C} : \{\pm 1\}^d \to \{\pm 1\}$.
for $\alpha \in \{\epsilon, 2\epsilon, 3\epsilon, \cdots 1-\epsilon, 1+200\epsilon\}$ **do**
  OptimizationResult $\leftarrow$ ELLIPSOIDALGORITHM $\left(1, \epsilon \cdot d^{-\frac{1}{2}\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}, \text{ORACLE}_{\alpha,d,\epsilon}\right)$.
  if OptimizationResult$\neq$FAIL **then**
    $P^{\text{GOOD}} =$ OptimizationResult
    $P^{\text{GOOD}}_{\text{TRIMMED}} \leftarrow$ representation of a function that takes input $\boldsymbol{x}$ and outputs the value

$$\begin{cases} P^{\text{GOOD}}(x) & \text{if } P^{\text{GOOD}}(x) \in [-1,+1] \\ 1 & \text{if } P^{\text{GOOD}}(x) > 1 \\ -1 & \text{if } P^{\text{GOOD}}(x) < -1 \end{cases}$$

    $P^{\text{GOOD}}_{\text{CORRECTED}} \leftarrow$ representation of a function that takes input $\boldsymbol{x}$ and returns the value

$$\text{HYPERCUBECORRECTOR}(x, P^{\text{GOOD}}_{\text{TRIMMED}}, r)$$

    $T \leftarrow \frac{200}{\epsilon^2} \log\left(\frac{20}{\epsilon}\right) \log(20d)$ i.i.d. pairs $(\boldsymbol{x}_i, f(\boldsymbol{x}_i))$, with $\boldsymbol{x}_i$ sampled uniformly from $\{-1,1\}^d$.
    ThresholdCandidates$\leftarrow \left\{\frac{1}{\epsilon} \text{ i.i.d. uniformly random elements in } [-1,1]\right\}$.
    $t^* := \arg\min_{t \in \text{ThresholdCandidates}} \left[\frac{1}{|T|} \sum_{\boldsymbol{x} \in T} \left[\left|\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t) - f(\boldsymbol{x})\right|\right]\right]$

    return representation of a function that takes input $\boldsymbol{x}$ and returns the value

$$\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t^*)$$

  end if
end for

Algorithm 16: Subroutine $\text{ORACLE}_{\alpha,d,\epsilon}(P)$

Given: $\epsilon, \alpha \in (0,1)$, degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial P over $\mathbb{R}^d$ with $\|P\|_2 \leq 1$, and uniform sample access to an unknown function $f : \{\pm 1\}^d \to \{\pm 1\}$.

Output: "Yes" or ("No", $Q_{\text{separator}}$), where $Q_{\text{separator}}$ is a degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial over $\mathbb{R}^d$.

$P_{\text{TRIMMED}} \leftarrow$ representation of a function that takes input $\boldsymbol{x}$ and outputs

$$\begin{cases} P(x) & \text{if } P(x) \in [-1,+1] \\ 1 & \text{if } P(x) > 1 \\ -1 & \text{if } P(x) < -1 \end{cases}.$$

$T \leftarrow$ set of $d^{\frac{C\sqrt{d}}{\epsilon} \log \frac{1}{\epsilon}}$ i.i.d. pairs $(\boldsymbol{x}_i, f(\boldsymbol{x}_i))$, with $\boldsymbol{x}_i$ sampled uniformly from $\{-1,1\}^d$ (for sufficiently large constant C).

$r \leftarrow$ string of $2^{C\sqrt{d}(\log d \cdot \log \frac{1}{\epsilon})^C}$ random i.i.d. bits (for sufficiently large constant C).

$M_{\text{separator}} \leftarrow$ representation of a function that takes input x and outputs

$$\begin{cases} 0 & \text{if } \text{HYPERCUBEMATCHING}(P_{\text{TRIMMED}}, \epsilon/4, r) \text{ does not match } x \text{ to any other vertex} \\ 1 & \text{if } \text{HYPERCUBEMATCHING}(P_{\text{TRIMMED}}, \epsilon/4, r) \text{ matches } x \text{ to some vertex } z, \text{ s.t. } z \preceq x \\ -1 & \text{if } \text{HYPERCUBEMATCHING}(P_{\text{TRIMMED}}, \epsilon/4, r) \text{ matches } x \text{ to some vertex } z, \text{ s.t. } z \succeq x \end{cases}$$

if $\frac{1}{|T|} \sum_{\boldsymbol{x} \in T} [M_{\text{separator}}(\boldsymbol{x}) \cdot P_{\text{TRIMMED}}(\boldsymbol{x})] > 5\epsilon$ **then**

  $Q_{\text{separator}} \leftarrow \sum_{S \subset [d]:\, |S| \leq \left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil} \left(\frac{1}{|T|} \sum_{\boldsymbol{x} \in T} [M_{\text{separator}}(\boldsymbol{x}) \cdot \chi_S(\boldsymbol{x})] \right) \chi_S$

  return ("No", $Q_{\text{separator}}$)

else if $\frac{1}{|T|} \sum_{\boldsymbol{x} \in T} [|f(\boldsymbol{x}) - P(\boldsymbol{x})|] > \alpha + 50\epsilon$ **then**

  $Q_{\text{separator}} \leftarrow \sum_{S \subset [d] |S| \leq \left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil} \left(\mathbb{E}_{\boldsymbol{x} \sim T} \left[\widehat{P}(S) \chi_S(\boldsymbol{x}) \text{sign}(P(\boldsymbol{x}) - f(\boldsymbol{x})) \right] \right) \chi_S$

  return ("No", $Q_{\text{separator}}$)

else

  return "Yes"

end if

Algorithm 17: MATCHVIOLATIONS$(P, f, \epsilon, \boldsymbol{r} = r_1 \circ \ldots \circ r_{\lceil \log 2/\epsilon \rceil})$

Given: Poset P and function $f : P \to [-1, 1]$ given as succinct representations, weight threshold ϵ, random seed $\boldsymbol{r} = r_1 \circ \ldots \circ r_{\lceil \log 2/\epsilon \rceil}$
Output: Succinct representation of a high-weight matching on the violating pairs of P w.r.t. f
if $\epsilon < 1/|P|$ **then**
  $M \leftarrow$ representation of the greedy algorithm that adds each edge (x, y) of $TC(P)$ in decreasing order of $f(x) - f(y)$.
else
  $t \leftarrow 2$
  $i \leftarrow 1$
  $M \leftarrow$ representation of a function computing the empty matching
  while $t > \epsilon/2$ **do**
    $P' \leftarrow$ representation of a function that takes input x and outputs FILTEREDGES$(TC(P), f, t, M, x)$
    $M \leftarrow$ representation of a function that takes input x and outputs $M(x)$ if $M(x) \neq \perp$, otherwise GHAFFARIMATCHING$(P', r_i, x))$
    $t \leftarrow t/2$
    $i \leftarrow i + 1$
  end while
end if

return M

Algorithm 18: Global view of sorting k-valued labels in a poset

Given: Poset P of height h, function $f : P \to [k]$
Output: monotone function $g : P \to [k]$
Let $i \leftarrow 0$
for $0 \leq i \leq \lceil \log k \rceil$ **do**
  Let f_i be the projection of f onto the i_{th} most significant bit of k, i.e. $f_i(x) = 1$ if the i_{th} bit of $f(x)$ is 1.
  Let P_i be the poset on the elements of P with the relation

$$x \prec_{P_i} y := x \prec_P y \text{ and } f_j(x) = f_j(y) \text{ for all } j < i.$$

  Let $\pi_i \leftarrow$ BOOLEANCORRECTOR(f_i, P_i).
  Let $f \leftarrow f\pi_i$.
end for

return f

queries, where Δ is the maximum number of predecessors or successors of any element in P, N is the number of vertices, and h is the length of the longest directed path.. It uses a random seed of length $\mathrm{poly}(\Delta \log N)$, and succeeds with probability $1 - N^{-10}$.

The following lemmas are used in the proof of correctness of our algorithm. Their proofs are deferred to the appendix.

Lemma 25 (Equivalence of k-valued and bitwise monotonicity)**.** *Let $f : P \to [k]$ be a function and f_i be the projection of f onto the i_{th} most significant bit of k, i.e. $f_i(x) = 1$ if the i_{th} bit of $f(x)$ is 1, for each $i \in [\lceil \log k \rceil]$. Let P_i be the poset on the elements of P with the relation*

$$x \prec_{P_i} y := x \prec_P y \text{ and } f_j(x) = f_j(y) \text{ for all } j < i.$$

Then f is monotone if and only if each f_i is monotone over the corresponding P_i.

Lemma 26 (Preservation of closeness to monotone functions)**.** *Let g be obtained from f by swapping the labels of a pair $x \prec_P y$ that violates monotonicity. Then for any monotone function m, $||g - m||_1 \leq ||f - m||_1$.*

The corollary follows from repeated application of Lemma 26 and the triangle inequality.

Corollary 12 (ℓ_1 error preservation)**.** *Let g be obtained from f by a series of swaps of label pairs that violate monotonicity in f. Then $||g - f||_1 \leq 2 \cdot \mathrm{dist}_1(f, \mathrm{mono})$.*

We also require a modification to the LCA claimed in Theorem 35 for correcting Boolean functions. That algorithm works by performing a sequence of label-swaps on pairs that violate monotonicity in the poset, then outputting the function value that ends up at the queried vertex x. It can instead track the swaps and output the identity of the vertex that x receives its final label from. The modified algorithm can be thought of as an LCA that gives query access to a label permutation.

Fact 10 (Poset sorting algorithm implicit in Chapter 4)**.** *Let P be a poset with N vertices such that every element has at most Δ predecessors and successors, and the longest directed path has length h. Let $f : P \to \{-1, 1\}$ be α-close to monotone in Hamming distance. There is an algorithm* BOOLEANCORRECTOR *that gives query access to a permutation π of P such that $f\pi$ is a monotone function and $\Pr_{x \sim P}[f(x) \neq (f\pi)(x)] \leq 2\alpha$. The LCA implementation of* BOOLEANCORRECTOR *uses $(\Delta \log N)^{O(\log h)}$ queries and running time, has a random seed of length $\mathrm{poly}(\Delta \log N)$, and succeeds with probability $1 - N^{-11}$.*

Here we present the LCA implementation of Algorithm 18.

Lemma 27 (Correctness and query complexity of Algorithm 19)**.** *With probability $1 - i \cdot N^{-11}$ over a random seed r of length $\mathrm{poly}(\Delta \log N)$, the algorithm k-*CORRECTOR*(x, P, f, i, r) gives query access to a function g that is monotone when truncated to the first i most significant bits. Its query complexity is $(\Delta \log N)^{O(i \log h + 1)}$, and $||g - f||_1 \leq 2\alpha$, where α is the ℓ_1 distance of f to the nearest monotone function.*

Algorithm 19: LCA implementation of Algorithm 18, k-CORRECTOR$(x, P, f, i, \boldsymbol{r})$

Given: Target vertex x, all-neighbors (immediate predecessor and successor) oracle for P, query access to $f : P \to [k]$, iteration number i, random seed $\boldsymbol{r} = r_1 \circ \ldots \circ r_i$.
Output: query access to function $g : P \to [k]$ which is monotone when truncated to the first i most significant bits.
if $i = 0$ **then**
  return $f(x)$
else
  $S \leftarrow$ the set of all predecessors and successors of x in P
  for $y \in S$ **do**
    Let $f'(y) \leftarrow k$-CORRECTOR$(y, P, f, i-1, r_1 \circ \ldots \circ r_{i-1})$.
  end for
  Let f'_i be defined as in Algorithm 18, and P'_i be similarly defined with respect to f'_i.
  Remove any y from S such that $f'_i(y) = f'_i(x)$ or y and x are incomparable in P'_i.
  Let $z \leftarrow$ BOOLEANCORRECTOR(x, P'_i, f'_i, r_i)

  return $f'(z)$
end if

Algorithm 20: HYPERCUBECORRECTOR$(f, \epsilon, \boldsymbol{r})$

Given: function $f : \{-1, 1\} \to [-1, 1]$ given as succinct representation, additive error parameter $\epsilon > 0$, random seed $\boldsymbol{r} = r_1 \circ \ldots \circ r_{\lceil \log 1/\epsilon \rceil}$.
Output: succinct representation of monotone function $g : \{-1, 1\} \to [-1, 1]$.
$P \leftarrow$ representation of a function that takes x and outputs TRUNCATEDCUBE(x, ϵ)
$f' \leftarrow$ representation of a function that takes x and outputs $\lfloor f(x)/\epsilon \rfloor$
$f'' \leftarrow$ representation of a function that takes x and outputs

$$\begin{cases} \epsilon \cdot k\text{-CORRECTOR}(x, P, f', \lceil \log(1/\epsilon) \rceil, \boldsymbol{r}) & -\sqrt{2d \log 2/\epsilon} \le |x| \le \sqrt{2d \log 2/\epsilon} \\ 1 & |x| \ge \sqrt{2d \log 2/\epsilon} \\ -1 & |x| \le -\sqrt{2d \log 2/\epsilon} \end{cases}$$

return f''

Proof. Fix the random seed $\boldsymbol{r}$ and assume all calls to BOOLEANCORRECTOR succeed with $\boldsymbol{r}$, then we proceed by induction. In the base case, f is certainly monotone when truncated to 0 bits and the algorithm makes only 1 query. In the inductive case, suppose the claim holds for $i-1$; in other words k-CORRECTOR$(y, P, f, i-1, r_1 \circ \ldots \circ r_{i-1})$ makes $(\Delta \log N)^{O((i-1)\log h+1)}$ queries and returns a function that is monotone in the first $i-1$ bits. Then when k-CORRECTOR is called with iteration number i, the function f'_j is monotone over P'_j for all $j < i$. BOOLEANCORRECTOR(x, P'_i, f'_i, r_i) returns a vertex to swap labels with x such that the resulting function is monotone in the i_{th} bit, over the poset P'_i. Then the function returned by k-CORRECTOR satisfies the conditions of Lemma 25 for the first i bits, so it must be monotone in the first i bits.

We now bound the failure probability and distance to f. The failure probability of BOOLEANCORRECTOR is N^{-11} and we call BOOLEANCORRECTOR on i different graphs, so by union bound the total failure probability is $\leq i \cdot N^{-11}$ as desired. The fact that $||g - f||_1 \leq 2\alpha$ follows from Corollary 12. □

We can now prove Theorem 33.

Theorem 33. *[Local monotonicity correction of real-valued functions] Let P be a poset with N elements, such that every element has at most Δ predecessors or successors and the longest directed path has length h. Let $f : P \to [-1, 1]$ be α-close to monotone in ℓ_1 distance. There is an LCA that makes queries to f and outputs queries to $g : P \to [-1, 1]$, such that g is monotone and $||f - g||_1 \leq 2\alpha + 3\epsilon$. The LCA makes $(\Delta \log N)^{O(\log h \log(1/\epsilon))}$ queries, uses a random seed of length $\mathrm{poly}(\Delta \log N)$, and succeeds with probability $1 - N^{-10}$.*

Proof of Theorem 33. Given some $\epsilon \in (0, 1/2)$, let $f_\epsilon(x) := \lfloor f(x)/\epsilon \rfloor$; certainly queries to f_ϵ can be simulated by queries to f. On input x, run k-CORRECTOR$(x, P, f_\epsilon, \lceil \log(2/\epsilon) \rceil, r)$ with a random seed $\boldsymbol{r}$ of length $\mathrm{poly}(\Delta \log N)$. By Lemma 27, this makes $(\Delta \log N)^{O(\log(1/\epsilon)\log h)}$ queries to f_ϵ and outputs $g_\epsilon(x)$, where g is monotone and $||g_\epsilon - f_\epsilon||_1 \leq 2 \cdot \mathrm{dist}_1(f_\epsilon, \mathrm{mono})$. Since f is α-close to some monotone function m, we have $\mathrm{dist}_1(f_\epsilon, \mathrm{mono}) \leq ||f_\epsilon - m/\epsilon||_1 \leq ||f/\epsilon - m/\epsilon||_1 + ||f/\epsilon - f_\epsilon||_1 \leq \alpha/\epsilon + 1$.

Return $g(x) := \epsilon \cdot g_\epsilon(x)$. Then

$$||g - f||_1 = ||\epsilon g_\epsilon - f||_1 \leq ||\epsilon g_\epsilon - \epsilon f_\epsilon||_1 + ||\epsilon f_\epsilon - f||_1 \leq$$
$$\leq 2\epsilon(\alpha/\epsilon + 1) + \epsilon \leq 2\alpha + 3\epsilon.$$

The failure probability is $N^{-11} \cdot \lceil \log(2/\epsilon) \rceil$ by Lemma 27, but we will assume that $\lceil \log(2/\epsilon) \rceil < N$. Otherwise, the allowed query complexity and running time would exceed Δ^N, which is $> \Delta N$ for any $\Delta, N > 1$. With $O(\Delta N)$ query complexity and running time, a trivial algorithm would suffice: one could solve the linear program with ΔN monotonicity constraints, minimizing $||g - f||_1$. Under our assumption, the failure probability is at most N^{-10}. □

Corollary 13 (Monotonizing a representation of a function on the Boolean cube)**.** *Let $f : \{-1, 1\}^d \to [-1, 1]$ be α-close to monotone in ℓ_1 distance, given as a succinct representation of size s_f. There*

is an algorithm that runs in time $2^{\tilde{O}(\sqrt{d}\log^{3/2}(1/\epsilon))} \cdot s_f$ *time and outputs a monotone function* g *such that* $||f-g||_1 \leq 2\alpha + 4\epsilon$*. The size of the representation of* g *is* $2^{\tilde{O}(\sqrt{d}\log^{3/2}(1/\epsilon))} \cdot s_f$*. The algorithm uses a random seed of length* $2^{\tilde{O}(\sqrt{d}\log(1/\epsilon))}$ *and succeeds with probability* $1 - 2^{-10d}$*.*

The proof of Corollary 13 is deferred to Section 5.6.6.

5.5 Analysis of the matching algorithm

In this section we give an algorithm for generating a succinct representation of a matching over the violated pairs of the hypercube whose weight is a constant factor of the distance to monotonicity. The core of the algorithm is an LCA for finding such a matching over the violated pairs of an arbitrary poset.

Lemma 28 (Equivalence of distance to monotonicity and maximum-weight matching)**.** *Let* W *be the total weight of the maximum-weight matching of the violation graph of* f*. Then* $\mathrm{dist}_1(f, \mathrm{mono}) = W/N$*.*

Proof. This proof is analogous to the proof of Lemma 3.1 of [BRY14a]; see Section 5.6.7. □

5.5.1 Details and correctness of MATCHVIOLATIONS

The algorithm MATCHVIOLATIONS given in Section 5.3 makes calls to an algorithm called FILTEREDGES, which removes vertices that have already been matched or are not incident to any heavy edges. We give the pseudocode for FILTEREDGES here.

Algorithm 21: LCA: FILTEREDGES(P, f, t, M, x)

Given: Poset P, function $f : P \to [-1, 1]$, and matching M given as succinct representations, weight threshold t, vertex x

Output: All neighbors of x in the graph of violation score $\geq t$ and not in M

return

$$\{y \in P(x) \mid M(y) = \perp \text{ and } [(x < y \text{ and } f(x) \geq f(y) + t) \text{ or } (x > y \text{ and } f(x) \leq f(y) - t)]\}$$

Lemma 29. *Let* P *be a poset with* N *vertices, and let* Δ *be an upper bound on the number of predecessors and successors of any vertex in* P*. Then the output of the LCA* MATCHVIOLATIONS(P, f, ϵ, r) *with a random seed* r *of length* $\mathrm{poly}(\Delta, \log N)$*, is a matching of weight at least* $N(\frac{1}{4}\mathrm{dist}_1(f, \mathrm{mono}) - \epsilon)$ *with probability at least* $1 - N^{-10}$*.*

Proof. This is a small modification to the standard greedy algorithm for high-weight matching; see Section 5.6.7. □

Lemma 30 (Running time and output size)**.** *Let $P, f, \epsilon, N, \Delta$, and r be as described in the lemma above. Let s_P be the size of the succinct representation of P, and s_f be the size of the succinct representation of f.*

Then MATCHVIOLATIONS(P, f, ϵ, r) *runs in time $(\Delta \log N)^{O(\log(1/\epsilon))}(s_P + s_f)$ and outputs a representation of size $(\Delta \log N)^{O(\log(1/\epsilon))}(s_P + s_f)$.*

Proof. If $\epsilon < 1/N$, then MATCHVIOLATIONS constructs and outputs a representation of the standard global greedy algorithm for 2-approximate maximum matching. The representation size of this algorithm is $O(\Delta N) \leq (\Delta \log N)^{O(\log(1/\epsilon))}$, and the running time of MATCHVIOLATIONS is polynomial in this representation size.

If $\epsilon \geq 1/N$, then by induction on the number of iterations i, we will show that the representation size of M at the start of iteration i is at most $(\Delta \log N)^{O(i)}(s_P + s_f)$. In the base case, we have an empty matching M which has constant representation size.

In the inductive case, suppose the claim holds at the start of iteration i. Then we set P' to be the function that applies FILTEREDGES to $TC(P)$. $TC(P)$ has size $O(\Delta \cdot s_P)$, as it makes $O(\Delta)$ calls to P. FILTEREDGES makes one call to $TC(P)$ and at most $O(\Delta)$ calls to M and f. It also has overhead of size $O(\log t) = O(\log(1/\epsilon)) = O(\log N)$. By the inductive hypothesis, the size of P' is then

$$O(\Delta) \cdot (\Delta \log N)^{O(i)}(s_P + s_f) + O(\log N) + O(\Delta \cdot s_P) \leq (\Delta \log N)^{O(i+1)}(s_P + s_f).$$

Then we set M to be the function that applies GHAFFARIMATCHING to P'. GHAFFARIMATCHING has constant overhead and makes $\mathrm{poly}(\Delta, \log N)$ queries to P'. Then the new size of M is $\mathrm{poly}(\Delta, \log N) \cdot (\Delta \log N)^{O(i)}(s_P + s_f) = (\Delta \log N)^{O(i+1)}(s_P + s_f)$.

The size bounds follow from the fact that there are $O(\log 1/\epsilon)$ iterations. The corresponding running time bound for MATCHVIOLATIONS comes from the fact that since it only constructs the succinct representations, its running time in each iteration is polynomial in the size of the representations it constructs.

□

Lemma 31. *With a random seed of length $2^{\tilde{O}(\sqrt{d}\log(1/\epsilon))}$, Algorithm 22 outputs a representation of a matching on the weighted violation graph $\mathrm{viol}(f)$, of weight at least $2^d \cdot (\frac{1}{4}\mathrm{dist}_1(f, \mathrm{mono}) - 4\epsilon)$, with probability at least $1 - 2^{-10d}$. The size of the representation is $2^{\tilde{O}(\sqrt{d}\log(1/\epsilon))} \cdot s_f$, where s_f is the size of the representation of f.*

Proof. HYPERCUBEMATCHING calls MATCHVIOLATIONS on the truncated hypercube, which has parameters $N < 2^d$ and $\Delta = 2^{O(\sqrt{d}\log d \log(1/\epsilon))}$. The size of the representation of TRUNCATEDCUBE is $O(d)$. So by Lemma 30, the running time and output size of HYPERCUBEMATCHING are $2^{O(\sqrt{d}\log d \log(1/\epsilon))} \cdot s_f$, and the random seed length is $2^{O(\sqrt{d}\log d \log(1/\epsilon))}$.

Algorithm 22: HYPERCUBEMATCHING$(f, \epsilon, \boldsymbol{r})$

Given: Function $f : \{-1,1\}^d \to [-1,1]$ given as succinct representation, weight threshold ϵ, random seed $\boldsymbol{r} = r_1 \circ \ldots \circ r_{\lceil \log 2/\epsilon \rceil}$
Output: Succinct representation of a high-weight matching on the violating pairs w.r.t. f
$P \leftarrow$ TRUNCATEDCUBE(d, ϵ)
$M \leftarrow$ representation of a function that takes x and outputs

$$\begin{cases} \text{MATCHVIOLATIONS}(P, f, \epsilon, \boldsymbol{r}) & -\sqrt{2d \log 2/\epsilon} \le |x| \le \sqrt{2d \log 2/\epsilon} \\ \perp & \text{otherwise} \end{cases}$$

return M

Let f' be the restriction of f to the truncated cube. Since f is bounded in $[-1,1]$ and the truncated cube covers all but an ϵ fraction of vertices, we have $\mathrm{dist}_1(f', \mathrm{mono}) \ge \mathrm{dist}_1(f, \mathrm{mono}) - 2\epsilon$. By Lemma 29, the weight of the matching is at least $(1-\epsilon) \cdot 2^d(\frac{1}{4}\mathrm{dist}_1(f', \mathrm{mono}) - \epsilon) \ge (1-\epsilon) \cdot 2^d(\frac{1}{4}\mathrm{dist}_1(f, \mathrm{mono}) - 3\epsilon/2) \ge 2^d(\frac{1}{4}\mathrm{dist}_1(f, \mathrm{mono}) - 4\epsilon)$.

□

5.6 Analysis of the agnostic learning algorithm

By inspecting algorithm MONOTONELEARNER (i.e. Algorithm 15 on page 187), we see immediately that the run-time is $2^{\tilde{O}(\sqrt{d}/\epsilon)}$. We proceed to argue that the algorithm indeed satisfies the guarantee of Theorem 32. First, we will need the following standard proposition.

Claim 10. *For any positive integers d and s, real $\epsilon, \delta \in (0,1)$, and any function $f : \{\pm 1\}^d \to [-1,1]$, let T be a collection of at least $d^{5s} \cdot \frac{100}{\epsilon^2} \ln \frac{1}{\epsilon} \ln \frac{1}{\delta}$ i.i.d. uniformly random elements of $\{\pm 1\}^d$. Then, with probability at least $1-\delta$*

$$\max_{\substack{\text{degree-}s\text{ polynomial } P \\ \text{with } \|P\|_2 \le 1}} \left| \|f - P\|_1 - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim T}[|f(\boldsymbol{x}) - P(\boldsymbol{x})|] \right| \le \epsilon,$$

Proof. See Section 5.6.8 for the proof of this proposition. □

Now, in the following lemma we prove that subroutine **Oracle**$_{\alpha,d,\epsilon}(P)$ (i.e. Algorithm 16 on page 188) satisfies some precise specifications with high probability. Informally, we show that **Oracle**$_{\alpha,d,\epsilon}(P)$ either

- Certifies that the polynomial P is both close to monotone in L_1 distance and has L_1 prediction error of $\alpha + O(\epsilon)$.
- Outputs a hyperplane separating P from all such polynomials.

Formally, we prove the following:

Lemma 32. *For sufficiently large constant C in Line 4 and Line 6 of procedure **Oracle**$_{\alpha,d,\epsilon}(P)$, sufficiently large integer d, any function $f : \{-1,1\}^d \to \{-1,1\}$, parameters $\epsilon, \alpha \in (0,1)$, and a degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial P satisfying $\|P\|_2 \leq 1$ the following is true. The procedure **Oracle**$_{\alpha,d,\epsilon}(P)$ runs in time $d^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ and will with probability at least $1 - \frac{1}{2^{5d}}$ conform to the following specification:*

1. *If **Oracle**$_{\alpha,d,\epsilon}(P)$ outputs "yes", then:*

   (a) *The function $P_{\text{TRIMMED}} = \begin{cases} 1 & \text{if } P(x) > 1, \\ -1 & \text{if } P(x) < -1, \\ P(\boldsymbol{x}) & \text{otherwise.} \end{cases}$ is 100ϵ-close to monotone in L_1 norm.*

   (b) *The L_1 distance between P and the function f is at most $\alpha + 100\epsilon$.*

2. *If **Oracle**$_{\alpha,d,\epsilon}(P)$ instead outputs ("No", $Q_{\text{separator}}$), where $Q_{\text{separator}}$ is a degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial over $\mathbb{R}^d$, then we have $\langle P', Q_{\text{separator}} \rangle < \langle P, Q_{\text{separator}} \rangle$ for any degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial P' with $\|P'\|_2 \leq 1$ that satisfies the following two conditions:*

   - *P' is ϵ-close in L_1 distance to some monotone function $f_{\text{monotone}} : \{\pm 1\}^d \to [-1,1]$ and*
   - *P' is $(\alpha+\epsilon)$-close in L_1 distance to the function f which we are trying to learn.*

*In particular, this implies that if P itself is ϵ-close in L_1 distance to some monotone function and is $(\alpha+\epsilon)$-close in L_1 distance to the function f, then **Oracle**$_{\alpha,d,\epsilon}(P)$ will say "yes" with probability at least $1 - \frac{1}{2^{10d}}$.*

Proof. We use the union bound to conclude that with probability at least $1 - \frac{1}{2^{5d}}$ all the following events hold:

(a) The LCA from Lemma 31 works as advertised and the weight W of the resulting matching satisfies

$$\frac{W}{2^d} \geq 0.1 \operatorname{dist}_1(P_{\text{TRIMMED}}, \text{mono}) - \epsilon.$$

Another way to write the same thing is

$$\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle \geq 0.1 \operatorname{dist}_1(P_{\text{TRIMMED}}, \text{mono}) - \epsilon. \tag{5.2}$$

From Lemma 31 it follows that this holds with probability at least $1 - \frac{1}{2^{10d}}$.

(b) The estimate of $\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle$ in Line 7 is indeed ϵ-close to the true value. From the standard Hoeffding bound, this holds with probability at least $1 - \frac{1}{2^{10n}}$.

(c) It is the case that

$$\left\| \sum_{\substack{S \subset [d]: \\ |S| \leq \left\lceil \frac{4 \cdot \sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}} \widehat{M_{\text{separator}}}(S) \chi_S - Q_{\text{separator}} \right\|_2 \leq \epsilon$$

Substituting the expression for $Q_{\text{separator}}$, and using the orthogonality of $\{\chi_S\}$ we see this is equivalent to

$$\sum_{S \subset [n]: |S| \leq \left\lceil \frac{4 \cdot \sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil} \underbrace{\left(\widehat{M_{\text{separator}}}(S) - \frac{1}{|T|} \sum_{\boldsymbol{x} \in T} [M_{\text{separator}}(\boldsymbol{x}) \cdot \chi_S(\boldsymbol{x})] \right)^2}_{\leq \epsilon d^{-\left\lceil \frac{4 \cdot \sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil} \text{ in absolute value w.p. } \geq \frac{1}{2^{10d}} \text{ via Hoeffding's bound}} \leq \epsilon$$

Overall, the above holds with probability at least $1 - \frac{1}{2^{9d}}$ by taking a Hoeffding bound for each individual summand and taking a union bound over them.

(d) The set $T \subset \{\pm 1\}^d$ is such that

$$\max_{\text{degree-}\left\lceil \frac{4 \cdot \sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil \text{ polynomial } P' \text{ over } \{\pm 1\}^d \text{with } \|P'\|_2 \leq 1} \left| \|f - P'\|_1 - \mathbb{E}_{(x, f(x)) \sim T} \left[|f(\boldsymbol{x}) - P'(\boldsymbol{x})| \right] \right| \leq \epsilon.$$

It follows from Claim 11 that this happens with probability at least to $1 - \frac{1}{2^{10d}}$.

Now, we argue that if these conditions indeed hold, then $\textbf{Oracle}_{\alpha,d,\epsilon}(P)$ will satisfy the specification given.

First, suppose $\textbf{Oracle}_{\alpha,d,\epsilon}(P)$ answered "yes". Then, since the estimate of $\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle$ in Line 7 is within ϵ of its true value, we have

$$\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle \leq 6\epsilon.$$

Now, since we are assuming the matching LCA from Lemma 31 works as advertised, this means that

$$6\epsilon \geq \langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle \geq 0.1 \cdot \text{dist}_1(P_{\text{TRIMMED}}, \text{mono}) - \epsilon$$

which can be rewritten as

$$\mathrm{dist}_1(P_{\text{TRIMMED}}, \mathrm{mono}) \leq 70\epsilon \leq 100\epsilon,$$

which is one of the two things we wanted to show. The other one was showing that the L_1 distance between P and the function f, which we are trying to learn, is at most $\alpha+100\epsilon$. Since the algorithm returned "yes", it has to be that in Line 11 we have

$$\mathbb{E}_{\boldsymbol{x}\sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] \leq \alpha + 50\epsilon.$$

From Section 5.6 it then follows that

$$\begin{aligned}\|f - P\|_1 &\leq \mathbb{E}_{\boldsymbol{x}\sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] + \epsilon \\ &\leq \alpha + 51\epsilon \leq \alpha + 100\epsilon,\end{aligned}$$

which is the other condition we wanted to show for the case when the oracle says "yes".

Now, assume the oracle outputs "no" along with some polynomial $Q_{\text{separator}}$ and let P' be a degree $\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial with $\|P'\|_2 \leq 1$ that satisfies the following two conditions[12]:

- P' is ϵ-close in L_1 distance to some monotone function $f_{\text{monotone}} : \{\pm 1\}^d \to [-1, 1]$ and
- P' is $(\alpha + \epsilon)$-close in L_1 distance to the function f which we are trying to learn.

Here, again, there are two cases. First, suppose we have the case where $Q_{\text{separator}}$ is generated from $M_{\text{separator}}$. We have that the oracle's estimate of $\langle M_{\text{separator}}, P_{\text{TRIMMED}}\rangle$ is at least 5ϵ, which means that $\langle M_{\text{separator}}, P_{\text{TRIMMED}}\rangle \geq 4\epsilon$. We know that P' is ϵ-close in L_1 distance to some monotone function $f_{\text{monotone}} : \{\pm 1\}^d \to [-1, 1]$. Since $M_{\text{separator}}$ is defined to be so for every matched pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ with $\boldsymbol{x}_i \prec \boldsymbol{y}_i$ we have $M_{\text{separator}}(\boldsymbol{x}_i) = 1$ and $M_{\text{separator}}(\boldsymbol{y}_i) = -1$ and is 0 otherwise, and for each such pair $f_{\text{monotone}}(\boldsymbol{x_i}) \leq f_{\text{monotone}}(\boldsymbol{y_i})$ we have $\langle M_{\text{separator}}, f_{\text{monotone}}\rangle \leq 0$. This allows us to conclude

$$\begin{aligned}0 \geq \langle M_{\text{separator}}, f_{\text{monotone}}\rangle &= \langle M_{\text{separator}}, P'\rangle + \langle M_{\text{separator}}, f_{\text{monotone}} - P'\rangle \geq \\ &\quad \langle M_{\text{separator}}, P'\rangle \\ &\quad - \left(\max_{x\in\{-1,1\}^n} |M_{\text{separator}}(x)|\right) \|f_{\text{monotone}} - P'\|_1 \\ &\geq \langle M_{\text{separator}}, P'\rangle - \epsilon,\end{aligned}$$

[12] If no polynomial satisfying these conditions exists, the statement we are seeking to prove holds vacuously.

which means

$$\begin{aligned}\epsilon \geq \langle M_{\text{separator}}, P' \rangle &= \left\langle \sum_{\substack{S \subset [n] \\ |S| \leq \left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}} \widehat{M_{\text{separator}}}(S) \left(\prod_{i \in S} x_i \right), P' \right\rangle \\ &= \langle Q_{\text{separator}}, P' \rangle - \\ &\quad - \left\| Q - \sum_{\substack{S \subset [n]: \\ |S| \leq \left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}} \widehat{M_{\text{separator}}}(S) \left(\prod_{i \in S} x_i \right) \right\|_2 \|P'\|_2 \\ &\geq \langle Q_{\text{separator}}, P' \rangle - \epsilon. \quad (5.3)\end{aligned}$$

On the other hand, the oracle's estimate of $\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle$ is at least 5ϵ, which means that it is the case that $\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle \geq 4\epsilon$. This allows us to conclude

$$\begin{aligned}4\epsilon \leq \overbrace{\langle M_{\text{separator}}, P_{\text{TRIMMED}} \rangle \leq \langle M_{\text{separator}}, P \rangle}^{\substack{\text{Trimming the values of a function} \\ \text{only decreases weights of violated edges.}}} &= \left\langle \sum_{S \subset [n]:\, |S| \leq \left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil} \widehat{M_{\text{separator}}}(S) \left(\prod_{i \in S} x_i \right), P \right\rangle \\ &\leq \langle Q_{\text{separator}}, P \rangle + \\ &\quad \left\| Q - \sum_{\substack{S \subset [n]: \\ |S| \leq \left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}} \widehat{M_{\text{separator}}}(S) \left(\prod_{i \in S} x_i \right) \right\|_2 \|P\|_2 \\ &\geq \langle Q_{\text{separator}}, P \rangle + \epsilon. \quad (5.4)\end{aligned}$$

Combining Equation 5.4 and Equation 5.3 we get

$$\langle Q_{\text{separator}}, P' \rangle \leq 2\epsilon < 3\epsilon \leq \langle Q_{\text{separator}}, P \rangle$$

as required.

Finally, we consider the case when $Q_{\text{separator}}$ is generated on Line 12. Since P' is $(\alpha + \epsilon)$-close

in L_1 distance to the function f, by Section 5.6 we have that

$$\alpha + \epsilon \leq \|f(\boldsymbol{x}) - P'(\boldsymbol{x})\|_1 \leq \mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - P'(\boldsymbol{x})|\right] - \epsilon,$$

which we can rewrite as $\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - P'(\boldsymbol{x})|\right] \leq \alpha + 2\epsilon$. At the same time, we have $\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] > \alpha + 50\epsilon$, which means that

$$\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] > \mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - P'(\boldsymbol{x})|\right].$$

Therefore, as the function mapping a polynomial H to the value $\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - H(\boldsymbol{x})|\right]$ is convex , it has to be the case that[13]

$$\begin{aligned}\left\langle P' - P, \sum_{\substack{S\subset[n]:\\ |S|\leq\left\lceil\frac{4\cdot\sqrt{n}}{\epsilon}\log\frac{4}{\epsilon}\right\rceil}} \left(\mathbb{E}_{\boldsymbol{x}\sim T}\left[\widehat{P}(S)\chi_S(\boldsymbol{x})\cdot \right.\right.\right. \\ \left.\left.\left. \cdot \operatorname{sign}(P(\boldsymbol{x}) - f(\boldsymbol{x}))\right]\right)\chi_S\right\rangle \\ = \left\langle P' - P, \nabla_H\left(\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|H(\boldsymbol{x}) - f(\boldsymbol{x})|\right]\right)\Big|_{H=P}\right\rangle \\ < 0.\end{aligned}$$

This implies that $\langle Q_{\text{separator}}, P'\rangle \leq \langle Q_{\text{sepatator}}, P\rangle$, which completes the proof. □

5.6.1 Finishing the proof of the Main Theorem (Theorem 32).

Recall that earlier by inspecting Algorithm 15 we concluded that this algorithm runs in time $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$. Here we use Lemma 32 to finish the proof of Theorem 32 by showing that with probability at least $1 - \frac{1}{2^d}$ the function $\operatorname{sign}(P^{\text{GOOD}}_{\text{TRIMMED}}(\boldsymbol{x}) - t^*)$ is monotone and is opt+$O(\epsilon)$-close to f (where opt is the distance of f to the closest monotone function).

We can further conclude that with probability at least $1 - \frac{1}{2^{3d}}$ the following events hold:

1. Every time an oracle $\textbf{Oracle}_{\alpha,d,\epsilon}$ is invoked (for various values of α), its behavior will conform to the specifications in Lemma 32.

2. The algorithm HypercubeCorrector from Corollary 13 used on line 11 works as advertised,

[13] To be fully precise, the expression above is a subgradient of the convex function mapping a polynomial H to $\mathbb{E}_{(\boldsymbol{x},f(\boldsymbol{x}))\sim T}\left[|f(\boldsymbol{x}) - H(\boldsymbol{x})|\right]$.

so the function $P^{\text{GOOD}}_{\text{CORRECTED}} : \{\pm 1\} \to [-1, 1]$ is monotone and we indeed have

$$\left\| P^{\text{GOOD}}_{\text{CORRECTED}} - P^{\text{GOOD}}_{\text{TRIMMED}} \right\|_1 \leq 10 \cdot \text{dist}_1(P^{\text{GOOD}}_{\text{TRIMMED}}, \text{mono}) + \epsilon. \quad (5.5)$$

3. In step (4), the function $\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t^*)$ satisfies the guarantee from Fact 11, i.e.

$$\Pr_{\boldsymbol{x} \sim \{\pm 1\}^d} \left[\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t^*) \neq f \right] \leq \frac{1}{2} \left\| P^{\text{GOOD}}_{\text{CORRECTED}} - f \right\|_1 + \epsilon \quad (5.6)$$

We argue that each of these events takes place with probability at least $1 - \frac{1}{2^{4d}}$:

- Note that the oracles $\textbf{Oracle}_{\alpha,d,\epsilon}$ for various values of α are invoked at most $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ times. Therefore, Lemma 32 tells us that for each of this invocations the algorithm $\textbf{Oracle}_{\alpha,d,\epsilon}$ conforms to its specification with probability at least $1 - \frac{1}{2^{5n}}$. Via union bound we see that event (1) holds with probability at least[14] $1 - \frac{1}{2^{4n}}$.
- Event (2) holds with probability at least $1 - \frac{1}{2^{4d}}$ via Corollary 13.
- Event (3) holds with probability at least $1 - \frac{1}{2^{4d}}$ via Fact 11

Via union bound, we see that with probability at least $1 - \frac{1}{2^{3d}}$ all these events hold, which we will assume for the rest of the proof.

Recall that opt stands for the distance of f to the closest monotone function. We first claim that the algorithm will break out of the loop in Line 13 for some value $\alpha^* \leq 2\text{opt} + 150\epsilon$, which we argue as follows: If $\alpha^* > 2\text{opt} + 150\epsilon$, then for some[15] $\alpha \in [2\text{opt}+100\epsilon, 2\text{opt}+150\epsilon]$ the ellipsoid algorithm failed to find some polynomial P on which $\textbf{Oracle}_{\alpha,n,\epsilon}$ returns "Yes". We claim that this is impossible. Indeed, let $\mathcal{C}_{\text{convex}}$ be the set consisting of degree-$\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomials P' with $\|P'\|_2 \leq 1$ that satisfies the following two conditions:

- P' is ϵ-close in L_1 distance to some monotone function $f_{\text{monotone}} : \{\pm 1\}^d \to [-1, 1]$, and
- P' is $(\alpha + \epsilon)$-close in L_1 distance to the function f which we are trying to learn.

We make the following observations:

[14] We assume that ϵ is such that $2^{0.1d}$ exceeds the number $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ of times that $\textbf{Oracle}_{\alpha,d,\epsilon}$ is invoked (for different values of α. Otherwise, the run-time budget is sufficient to store entire truth-tables of functions over $\{-1, 1\}^d$ and statement in Algorithm 20 is achieved by the trivial algorithm that uses a linear program to fit the best montone real-valued function and then rounds it to be $\{-1, 1\}$-valued. See Section 5.6.4 for further details.

[15] Note that opt $\leq 1/2$, because the function f is at least $1/2$-close to either the all-ones or all-zeroes functions, which are both monotone. Therefore some value of α in the range $[2\text{opt}+100\epsilon, 2\text{opt}+150\epsilon]$ is necessarily considered by the algorithm as it is trying all values $\alpha = \epsilon, 2\epsilon, 3\epsilon, \cdots 1 - \epsilon, 1 + 200\epsilon$.

- The set $\mathcal{C}_{\text{convex}}$ is a convex set, because (a) the set of all monotone functions $f_{\text{monotone}} : \{\pm 1\}^d \to [-1, 1]$ is convex, (b) the set of points $(\alpha + \epsilon)$-close in L_1 distance to some specific convex set is itself convex, and (c) the intersection of two convex sets is a convex set (in this case one convex set is the set functions $\{\pm 1\}^n \to [-1, 1]$ that are $(\alpha + \epsilon)$-close in L_1 distance a monotone functions and the other convex set is is the set of all degree-$\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomials with with $\|P'\|_2 \leq 1$).

- The set $\mathcal{C}_{\text{convex}}$ contains an L_2 ball of radius at least $\epsilon \cdot n^{-\frac{1}{2}\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}$. In other words, in $\mathcal{C}_{\text{convex}}$ there is some degree $\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial P_0 such that any degree-$\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial P' that is ϵ-close to P_0 in L_2 norm is also in $\mathcal{C}_{\text{convex}}$. Let $f_{\text{monotone, optimal}} : \{\pm 1\}^n \to \{\pm 1\}$ be the monotone function for which it is the case that $\Pr_{\boldsymbol{x}\sim\{\pm 1\}^n}\left[f_{\text{monotone, optimal}}(\boldsymbol{x}) \neq f(\boldsymbol{x})\right] =$ opt, and let P_0 be a degree-$\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ polynomial that is ϵ-close to $f_{\text{monotone, optimal}}$ in L_1 norm (such polynomial has to exist by Fact 7). Then, P_0 is $(2\text{opt} + \epsilon)$-close to f in L_1 norm and ϵ-close to monotone in L_1 norm. In other words, the set $\mathcal{C}_{\text{convex}}$ contains an L_1-ball of radius ϵ. Via the standard inequality between the L_1 and L_2 norms, in d dimensions every L_1 ball or radius ϵ contains an L_2 ball of radius at most $\epsilon/\sqrt{d}$. Our claim follows, since the space of degree-$\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil$ over $\mathbb{R}^d$ has dimension at most $n^{\left\lceil \frac{4\cdot\sqrt{n}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}$.

- Since the procedure $\text{Oracle}_{\alpha,d,\epsilon}$ is assumed to satisfy the specifications given in Lemma 32 and for this specific value of α it never gave the response "yes", then for every query P to $\text{Oracle}_{\alpha,d,\epsilon}$, the oracle returned some halfspace that separates P from the convex set $\mathcal{C}_{\text{convex}}$.

From Fact 8 we know that under these conditions the ellipsoid algorithm will necessarily in time $\text{poly}\left(d^{\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil}, \log\left(R/r\right)\right) = d^{O\left(\left\lceil \frac{4\cdot\sqrt{d}}{\epsilon} \log \frac{4}{\epsilon} \right\rceil\right)}$ find some polynomial P that is in $\mathcal{C}_{\text{convex}}$. For this particular polynomial, the specifications in Lemma 32 require the oracle $\text{Oracle}_{\alpha,d,\epsilon}$ to give a response "yes", which gives us a contradiction. Thus, the function $P^{\text{GOOD}}_{\text{TRIMMED}}$ will be $O(\epsilon)$-close to monotone in L_1 norm and will satisfy $\left\|P^{\text{GOOD}}_{\text{TRIMMED}} - f\right\|_1 \leq 2\text{opt} + O(\epsilon)$. Combining this with Equation (5.5) yields

$$\left\|P^{\text{GOOD}}_{\text{CORRECTED}} - f\right\|_1 \leq 2\text{opt} + O(\epsilon) + \left\|P^{\text{GOOD}}_{\text{TRIMMED}} - P^{\text{GOOD}}_{\text{CORRECTED}}\right\|_1 = 2\text{opt} + O(\epsilon).$$

We know that $\left\|P^{\text{GOOD}}_{\text{TRIMMED}} - P^{\text{GOOD}}_{\text{CORRECTED}}\right\|_1 \leq O(\epsilon)$ because $P^{\text{GOOD}}_{\text{TRIMMED}}$ is $O(\epsilon)$-close to monotone by Equation (5.5). Now, combining the inequality above with Equation 5.6 gives us

$$\Pr_{\boldsymbol{x}\sim\{\pm 1\}^n}\left[\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t^*) \neq f\right] \leq \frac{1}{2}\left\|P^{\text{GOOD}}_{\text{CORRECTED}} - f\right\|_1 + \epsilon \leq \text{opt} + O(\epsilon).$$

Finally, we see that since the function $P^{\text{GOOD}}_{\text{CORRECTED}}\{\pm 1\}^d \to [-1,+1]$ is monotone we have that the $\{\pm 1\}$-valued function $\text{sign}(P^{\text{GOOD}}_{\text{CORRECTED}}(\boldsymbol{x}) - t^*)$ is also monotone, which finishes our argument.

5.6.2 Rounding of real-valued functions to Boolean.

Fact 11. *Suppose we have two functions $g : \{\pm 1\}^d \to \mathbb{R}$ and $f : \{\pm 1\}^d \to \{\pm 1\}$. Let T be a set of at least $\frac{40}{\epsilon^2}\log\left(\frac{20}{\epsilon\delta}\log\frac{1}{\delta}\right)$ i.i.d. uniformly random elements of $\{-1,1\}^d$, and let ThresholdCandidates $\subset [-1,1]$ be a set of $\frac{20}{\epsilon}\log\frac{1}{\delta}$ i.i.d. uniformly random elements of $[-1,1]$. Let*

$$t^* := \operatorname*{arg\,min}_{t \in \textit{ThresholdCandidates}} \frac{1}{|T|}\sum_{\boldsymbol{x}\in T} |\textit{sign}(g(\boldsymbol{x}) - t) - f(\boldsymbol{x})|$$

Then, with probability at least $1-\delta$ it is the case that

$$\Pr_{\boldsymbol{x}\sim\{\pm 1\}^d}[\textit{sign}(g(\boldsymbol{x}) - t^*) \neq f] \leq \tfrac{1}{2}\|f-g\|_1 + \epsilon$$

Proof. We get that

$$\mathbb{E}_{t\sim[-1,1]}\left[\mathbb{E}_{\boldsymbol{x}\sim\{\pm 1\}^n}[|\text{sign}(g(\boldsymbol{x}) - t) - f(\boldsymbol{x})|]\right] \leq \|f-g\|_1$$

directly via linearity of expectation. Now, the random variable $\mathbb{E}_{\boldsymbol{x}\sim\{\pm 1\}^d}[|\text{sign}(g(\boldsymbol{x}) - t) - f(\boldsymbol{x})|]$ (with randomness taken over t) is always in $[0,2]$ and has some expectation $E \in [0,2]$ which is at most $\|f-g\|_1$. By Markov's inequality, we have

$$\Pr_{t\sim[-1,1]}[\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{\pm 1\}^n}[|\text{sign}(g(\boldsymbol{x}) - t) - g(\boldsymbol{x})|] \geq E + \epsilon/2] \leq \frac{E}{E+\epsilon/2} \leq \frac{2}{2+\epsilon/2} \leq 1 - \frac{\epsilon}{4}.$$

Since the set ThresholdCandidates consists of $\frac{20}{\epsilon}\log\frac{1}{\delta}$ i.i.d. uniform elements in $[-1,1]$, then with probability $1-\delta$ or more, some t in ThresholdCandidates will satisfy the condition that $\mathbb{E}_{\boldsymbol{x}\sim\{\pm 1\}^d}[|\text{sign}(g(\boldsymbol{x}) - t) - g(\boldsymbol{x})|]$ is in $[0, E+\epsilon/2]$.

Finally, from the Hoeffding bound and union bound we observe that with probability at least $1 - \frac{\delta}{2}$ it is the case that

$$\max_{t\in\text{ThresholdCandidates}}\left|\frac{1}{|T|}\sum_{\boldsymbol{x}\in T}|\text{sign}(g(\boldsymbol{x}) - t) - f(\boldsymbol{x})| - E_{\boldsymbol{x}\sim\{-1,1\}^n}|\text{sign}(g(\boldsymbol{x}) - t) - f(\boldsymbol{x})|\right| \leq \frac{\epsilon}{4}.$$

Overall, we see that with probability at least $1-\delta$ it is the case that

$$\begin{aligned}\Pr_{\boldsymbol{x}\sim\{\pm1\}^n}[\text{sign}(g(\boldsymbol{x})-t^*)\neq f] &\\ \leq \frac{1}{|T|}\sum_{\boldsymbol{x}\in T}|\text{sign}(g(\boldsymbol{x})-t^*)-f(\boldsymbol{x})|+\frac{\epsilon}{4} &\\ \leq \tfrac{1}{2}\left\|f-g\right\|_1+\epsilon &\end{aligned}$$

This finishes the proof. □

5.6.3 Agnostic learning algorithms handling randomized labels.

It is customary in the agnostic learning literature to consider a setting that is slightly more general than the one in Theorem 32. Specifically, one is given pairs of i.i.d. elements $\{(x_i, y_i)\}$ from a distribution D_{pairs}, where the distribution of each x_i by itself is uniform. The aim here is to output an efficiently-evaluable succinct representation of a function g for which

$$\begin{aligned}\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim D_{\text{pairs}}}[g(\boldsymbol{x})\neq\boldsymbol{y}] &\\ \leq \min_{\substack{\text{monotone}\, f_{\text{mon}}:\\ \{-1,1\}^n\to\{-1,1\}}} \Pr_{(\boldsymbol{x},\boldsymbol{y})\sim D_{\text{pairs}}}[f_{\text{mon}}(\boldsymbol{x})\neq\boldsymbol{y}]+O(\epsilon). &\quad (5.7)\end{aligned}$$

The only difference between this setting and the one in Theorem 32 is that here the label y doesn't have to be a function of example x; it is possible to receive the same example x twice accompanied by different labels. Here we argue that Theorem 32 extends directly into this slightly more general setting. Formally, we show that

Theorem 36. *For all sufficiently large integers n the following holds. There is an algorithm that runs in time $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ and given i.i.d. samples of pairs $\{(x_i, y_i)\}$ from a distribution D_{pairs}, where the marginal distribution over x is uniform, does the following. With probability at least $1-\frac{1}{2^{0.5d}}$ the algorithm outputs a representation of a monotone function $g:\{\pm1\}^d\to\{\pm1\}$ of size $2^{\tilde{O}\left(\frac{\sqrt{d}}{\epsilon}\right)}$ that satisfies Equation* (5.7).

5.6.4 Case 1: ϵ is very small.

We will consider two cases. First of all, suppose ϵ is so small that the run-time of the algorithm in Theorem 32 exceeds $2^{0.1d}$. In this case, the following algorithm runs in time $\text{poly}(2^d, 1/\epsilon)$ and outputs and efficiently-evaluable succinct representation of a function g for which Equation (5.7) holds:

1. Draw two sets T_1 and T_2, each of $100d^5\cdot 2^d/\epsilon^2$ example-label pairs from D_{pairs}.
2. For each $x\in\{-1,1\}^n$ let $h(x)$ be $\frac{1}{|(x_i,y_i)\in T_1 \text{ s.t. } x_i=x|}\sum_{(x_i,y_i)\in T_1 \text{ s.t. } x_i=x} y_i$.

3. Via a size-$2^{O(n)}$ linear program, find the monotone function $q : \{-1,1\}^d \to [-1,1]$ that is closest to h is ℓ_1 distance.

4. Output the function g defined so $g(x) := \text{sign}(q(x) - t^*)$, where t^* is obtained as in Fact 11 using the samples in T_2.

The function g we output above with high probability satisfies Theorem 32 for the following reason. First of all, via the standard coupon-collector argument with probability at least $1 - \frac{1}{2^{5d}}$ for every $x \in \{-1,1\}^n$ there will be at least $10^2/\epsilon^2$ elements in (x_i, y_i) in T for which $x_i = x$. Using the Hoeffding bound and the union bound, we see that with probability at least $1 - \frac{1}{2^{2n}}$ we have

$$\left| h(x) - \mathbb{E}_{(\boldsymbol{x}', \boldsymbol{y}') \sim D_{\text{pairs}}} \left[\boldsymbol{y}' \middle| \boldsymbol{x}' = x \right] \right| \leq \frac{\epsilon}{2}. \tag{5.8}$$

Now, from steps (3) and (4) we have

$$\frac{\|h - g\|_1}{2} \leq \tfrac{1}{2}\text{dist}_1(h, \text{mono}) + \epsilon. \tag{5.9}$$

Therefore, we can combine Equation (5.8) and Equation (5.9) to obtain

$$\Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim D_{\text{pairs}}}[g(\boldsymbol{x}) \neq \boldsymbol{y}] \leq \min_{\substack{\text{monotone } f_{\text{mon}}: \\ \{-1,1\}^n \to \{-1,1\}}} \Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim D_{\text{pairs}}}[f_{\text{mon}}(\boldsymbol{x}) \neq \boldsymbol{y}] + O(\epsilon), \tag{5.10}$$

which finishes the proof for this case.

5.6.5 Case 2: ϵ is not too small.

Now, we proceed to the other case when ϵ is not too small and the algorithm in Theorem 32 runs in time at most $2^{0.1n}$ (and therefore uses at most $2^{0.1d}$ samples). In this case, we claim that simply running the algorithm in Theorem 32 will give an efficiently evaluable succinct description of a function g that satisfies the guarantee in Equation (5.7).

We now proceed to show that the guarantee in Equation (5.7) will indeed be achieved. Define a random function $f_{\text{random}} : \{-1,1\}^n \to \{-1,1\}$, so for all $x \in \{-1,1\}^n$ the value $f_{\text{random}}(x)$ is chosen independently such that $f_{\text{random}}(x) = 1$ with probability $\Pr_{(\boldsymbol{x}', \boldsymbol{y}') \sim D_{\text{pairs}}}[\boldsymbol{y}' = 1 \mid \boldsymbol{x}' = x]$ and $f_{\text{random}}(x) = -1$ with probability $\Pr_{(\boldsymbol{x}', \boldsymbol{y}') \sim D_{\text{pairs}}}[\boldsymbol{y}' = -1 \mid \boldsymbol{x}' = x]$. Consider the following two scenarios:

- **Scenario I:** The samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$ given to the algorithm from Theorem 32 are indeed i.i.d. samples coming from D_{pairs}.

- **Scenario II:** The samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$ given to the algorithm from Theorem 32 are sampled as follows: (i) $\boldsymbol{x}_i$ are i.i.d. uniform from $\{-1,1\}^d$ (ii) $\boldsymbol{y}_i = f_{\text{random}}(\boldsymbol{x}_i)$.

First we argue that in Scenario II with probability at least $1 - \frac{2}{2^d}$ the function g given by the algorithm from Theorem 32 satisfies Equation (5.7), (here the probability is over the choice of f_{random}, choice of the samples, and the randomness of the algorithm itself). Indeed, let f^*_{mon} be the function that minimizes the right side of Equation (5.7). From the Hoeffding's bound, it follows that with probability at least[16] $1 - \frac{1}{2^d}$ over the choice of f_{random} it is the case that

$$\left| \Pr_{\boldsymbol{x} \sim \{-1,1\}^n} [f_{\text{random}}(\boldsymbol{x}) \neq f^*_{\text{mon}}(\boldsymbol{x})] - \Pr_{(\boldsymbol{x},\boldsymbol{y}) \sim D_{\text{pairs}}} [f^*_{\text{mon}}(\boldsymbol{x}) \neq \boldsymbol{y}] \right| \leq \epsilon. \quad (5.11)$$

Now, Theorem 32 implies that with probability at least $1 - \frac{1}{2^d}$

$$\Pr_{\boldsymbol{x} \sim \{-1,1\}^n} [g(\boldsymbol{x}) \neq f_{\text{random}}(\boldsymbol{x})] \leq \text{dist}_0(f_{\text{random}}, \text{mono}) + O(\epsilon) \leq \Pr_{\boldsymbol{x} \sim \{-1,1\}^n} [f^*_{\text{mon}}(\boldsymbol{x}) \neq f_{\text{random}}(\boldsymbol{x})] + O(\epsilon). \quad (5.12)$$

Combining Equations 5.11 and 5.12 we we see that with probability at least $1 - \frac{2}{2^d}$, the function g given by the algorithm from Theorem 32 satisfies Equation (5.7) in Scenario II.

Finally, we argue that Equation (5.7) will be satisfied also in Scenario I with probability at least $1 - \frac{1}{2^{0.5d}}$ for sufficiently large d. Conditioned on the absence of sample pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ with $\boldsymbol{x}_i = \boldsymbol{x}_j$, the distributions over samples in Scenario I and Scenario II are the same, Hence it suffices to argue that the collision probability is low, given that the value of ϵ is such that the algorithm from Theorem 32 uses at most $2^{0.1d}$ samples. By taking a union bound over all pairs of samples, we bound the probability of such collision by $\frac{2^{0.2d}}{2^d} = 2^{-0.8n}$. Thus, information-theoretically, any algorithm can distinguish between Scenario I and Scenario II with an advantage of only at most $2^{-0.8d}$. In particular, this is true of the algorithm that checks whether Equation (5.7) applies. Thus, indeed Equation (5.7) will be satisfied also in Scenario I with probability at least $1 - \frac{2}{2^d} - \frac{1}{2^{0.8d}} \geq 1 - \frac{1}{2^{0.5d}}$, which finishes the proof of Theorem 36.

5.6.6 Proofs deferred from Section 5.4

Proof of Lemma 25. Let x and y be comparable elements of P; w.l.o.g. $x \prec_P y$. It is sufficient to show that $f(x) > f(y)$ if and only if there is some i for which $x \prec_{P_i} y$ and $f_i(x) > f_i(y)$. We claim that this i is the most significant bit in which $f(x)$ and $f(y)$ differ. It is certainly true that $f(x) > f(y)$ if and only if $f_i(x) > f_i(y)$ for this i, and since $f_j(x) = f_j(y)$ for all $j < i$ by the choice of i, we have $x \prec_{P_i} y$ as well. □

[16] Here we used that $\epsilon \geq \frac{1}{\sqrt{d} \operatorname{poly} \log d}$, because otherwise ϵ would be too small and we would be in the other case when the run-time of the algorithm in Theorem 32 exceeds $2^{0.1d}$. Also, we note that a much stronger bound can be deduced from the Hoeffding bound, but we only need a bound of $1 - \frac{1}{2^d}$.

Proof of Lemma 26. Since m is monotone, certainly $m(x) \leq m(y)$, and since f violates monotonicity on this pair, certainly $f(x) \geq f(y)$ (and therefore $g(y) \geq g(x)$). We will examine the contribution of x and y to each of $||f - m||_1$ and $||g - m||_1$. We have the following cases:

- $f(y) \leq f(x) \leq m(x) \leq m(y)$: then

$$\begin{aligned}&|m(x) - f(x)| + |m(y) - f(y)|\\ =&m(x) + m(y) - (f(x) + f(y))\\ =&m(x) + m(y) - (g(x) + g(y))\\ =&|m(x) - g(x)| + |m(y) - g(y)|.\end{aligned}$$

  The distance of this pair does not change. The case of $m(x) \leq m(y) \leq f(x) \leq f(y)$ is symmetric.

- $f(y) \leq m(x) \leq m(y) \leq f(x)$: then

$$\begin{aligned}&|m(x) - f(x)| + |m(y) - f(y)|\\ =&(f(x) - m(x)) + (m(y) - f(y))\\ \geq&(f(x) - m(y)) + (m(x) - f(y))\\ =&|g(y) - m(y)| + |g(x) - m(x)|.\end{aligned}$$

  The distance of this pair does not increase. The case of $m(x) \leq f(y) \leq f(x) \leq m(y)$ is symmetric.

- $f(y) \leq m(x) \leq f(x) \leq m(y)$: then

$$\begin{aligned}&|m(x) - f(x)| + |m(y) - f(y)|\\ =&(f(x) - m(x)) + (m(y) - f(y))\\ \geq&(m(x) - f(y)) + (m(y) - f(x))\\ =&|g(x) - m(x)| + |g(y) - m(y)|.\end{aligned}$$

  The distance of this pair does not increase. The case of $m(x) \leq f(y) \leq m(y) \leq f(x)$ is symmetric.

□

Proof of Corollary 13. Let $f : \{-1, 1\}^d \to [-1, 1]$ be α-close to monotone in ℓ_1 distance. We call the algorithm HYPERCUBECORRECTOR(f, ϵ, r) with a random seed r of length $2^{O(\sqrt{d \log(1/\epsilon)} \log n)}$. First we set the poset to be the truncated cube of width $\sqrt{2d \log 2/\epsilon}$, which is a poset such that every element has at most $2^{O(\sqrt{d \log(1/\epsilon)} \log d)}$ predecessors and successors. The representation of this poset (not its transitive closure) has size $\mathrm{poly}(d, \log(1/\epsilon))$. Then we set f' to be a function that discretizes f to $2/\epsilon$ possible values. This representation has size $O(s_f/\epsilon)$. Then we set f'' to be a function that computes the Hamming weight of x, then either calls k-CORRECTOR or outputs a constant.

So its size is the size of the k-CORRECTOR representation times some overhead that is polynomial in d and $1/\epsilon$. Since the Δ parameter for the truncated cube is $2^{O(\sqrt{d \log(1/\epsilon)} \log d)}$, the h parameter is $O(\sqrt{d})$, and the N parameter is $< 2^d$, the worst-case running time and query complexity of this instance of k-CORRECTOR is $2^{O(\sqrt{d} \log d \log^{3/2}(1/\epsilon))}$ by Lemma 27. Thus the representation size of the k-CORRECTOR instance is $2^{\tilde{O}(\sqrt{d} \log^{3/2}(1/\epsilon))}$, and so the representation size of f'' is $2^{\tilde{O}(\sqrt{d} \log^{3/2}(1/\epsilon))} \cdot s_f$. With the random seed of length $2^{O(\sqrt{d \log(1/\epsilon)} \log d)} = \mathrm{poly}(\Delta \log N)$, k-CORRECTOR succeeds with probability $N^{-10} \leq 2^{-10d}$.

□

5.6.7 Proofs deferred from Section 5.5

Proof of Lemma 28. The proof of $\mathrm{dist}_1(f, \mathrm{mono}) \geq W/N$ is straightforward; for any edge (x, y), $x \prec y$ in the matching, any monotone function must have $g(y) \geq g(x)$ and thus $(f(x) - g(x)) + (g(y) - f(y)) \geq f(x) - f(y)$. So the contribution of x and y to the ℓ_1 distance is at least the weight of (x, y).

For the other direction, we give a proof exactly analogous to the max-weight matching characterization of distance to the class of Lipschitz functions, presented in [BRY14a]. Let g be the closest monotone function to f in ℓ_1-distance. We will partition the vertices of the cube into three classes: $V_> := \{x \mid f(x) > g(x)\}$, $V_< := \{x \mid f(x) < g(x)\}$, and $V_= := \{x \mid f(x) = g(x)\}$. We will duplicate the vertices of $V_=$ and group one copy with $V_>$ and one copy with $V_<$, to form vertex sets $V_\geq$ and $V_\leq$. The duplicated copies of x will be denoted $x_\geq$ and $x_\leq$. We define the bipartite graph $B_{f,g}$ to be the graph on $V_\geq \times V_\leq$ with an edge (x, y) if $x \prec y$ and $g(x) = g(y)$. The weight of the edge (x, y) is the same as it is in $\mathrm{viol}(f)$; it is just $f(x) - f(y)$. Intuitively, a matching in $B_{f,g}$ will represent a set of edges along which some a minimal amount of label mass is transferred to correct monotonicity. First, we claim that $B_{f,g}$ has a matching which matches every vertex in $V_> \cup V_<$. This will follow from Hall's marriage theorem if we can show that for every $A \subseteq V_>$ or $A \subseteq V_<$, we have $|A| \leq |N(A)|$.

Suppose for contradiction that the marriage condition is false, and without loss of generality let A be the largest subset of $V_>$ for which $|A| > |N(A)|$. We would like to claim that for any $x \in A \cup N(A)$ and $y \notin A \cup N(A)$, if $x \prec y$ then $g(x) < g(y)$. We consider four possible cases:

a) If $x \in A$, $y \in V_>$, $x \prec y$, and $g(x) = g(y)$, then $y \in A$ as well, by the choice of A to be the largest set that fails the marriage condition. This is because $N(y) \subseteq N(x)$: any neighbor z of y must have $g(z) = g(y) = g(x)$, have $x \prec y \prec z$, and be in $V_\leq$, which makes it a neighbor of x.

b) If $x \in N(A)$, $y \in V_\leq$, $x \prec y$, and $g(x) = g(y)$, then $g(y) = g(x) = g(z)$ and $z \prec x \prec y$ for some $z \in A$, so $y \in N(A)$.

c) If $x \in A$, $y \in V_\leq$, $x \prec y$, and $g(x) = g(y)$, then $y \in N(A)$.

d) If $x \in N(A)$, $y \in V_>$, $x \prec y$, and $g(x) = g(y)$, then $g(y) = g(x) = g(z)$ and $z \prec x \prec y$ for some $z \in A$, so as in case (a) we have $N(y) \subseteq N(z)$ and therefore $y \in A$.

We have shown that for any $x \in A \cup N(A)$ and $y \notin A \cup N(A)$, if $x \prec y$ then $g(x) < g(y)$. Then there is some $\delta > 0$ for which $g(x)$ can be increased by δ for every $x \in A \cup N(A)$ without breaking monotonicity. This decreases $||f - g||_1$ by $\delta(|A| - N(A)|) > 0$, which contradicts the assumption that g is the closest monotone function.

Having proven that $B_{f,g}$ contains a matching M' on all vertices in $V_> \cup V_<$, we will now show that its weight is equal to $N||f - g||_1$, using the fact that $g(x) = g(y)$ for all $(x, y) \in M'$:

$$\sum_{(x,y)\in M'} f(x) - f(y) = \sum_{(x,y)\in M'} f(x) - g(x) + g(y) - f(y) = \sum_{x \in V_> \cup V_<} |f(x) - g(x)| = N||f - g||_1.$$

We will now find a matching M in $\mathrm{viol}(f)$ of equal weight. First replace each $x_\leq$ and $x_\geq$ with x, obtaining an edge set in $\mathrm{viol}(f)$ of equal weight that is not necessarily a matching, but is a set of disjoint paths. We replace each path with the edge between its endpoints; i.e. if there is some pair of edges $(y, x_\leq)$ and $(x_\geq, z)$, then we know that $y \prec x \prec z$ and $f(y) - f(z) = ((f(y) - f(x) + (f(x) - f(z))$, so the matching edge (y, z) has weight equal to the total weight of the path it replaces. Then M is a matching in $\mathrm{viol}(f)$ of weight equal to $N||f - g||_1$, which is equal to $N \cdot \mathrm{dist}_1(f, \mathrm{mono})$. □

Proof of Lemma 29. Fix the random seed r and assume all calls to the algorithm of [Gha22] using r succeed. Let M' be a maximum-weight matching over $\mathrm{viol}(f)$, and let M be a matching returned by MATCHVIOLATIONS. We will use M to refer to the matching and its succinct representation interchangeably. For each edge $e \in M'$, let w_e be the weight of e (i.e. the violation score of its endpoints), and δ_e be the total weight of edges in $M \setminus M'$ that share an endpoint with e.

First we show by induction that at the start of each iteration i, M is maximal over the subgraph of $TC(P)$ induced by edges of weight greater than $2^{-(i-1)}$. In the base case, M is initialized to be the empty matching, which is maximal on the edges of weight > 2, as there are no such edges. In the inductive case, we assume the invariant is still true at the start of iteration i. Then when FILTEREDGES (Algorithm 21) is called in iteration $i + 1$, the vertices removed are exactly those that are either already in M, or not incident to any edges of weight greater than $t = 2^{-i}$. Then by the maximality of the matching computed by GHAFFARIMATCHING on the filtered subgraph, any edge not in that matching must satisfy one of the following criteria:

- it has weight at most 2^{-i},
- it has an endpoint in M,
- it shares an endpoint with another edge in GHAFFARIMATCHING.

So after the new edges of in GHAFFARIMATCHING are added to M, M is maximal over the 2^{-i}-heavy edges as desired.

Now we claim that $\delta_e \geq w_e/2$ for any edge $e \in M' \setminus M$ of weight at least ϵ. This is because after the first round for which $t < w_e$, M' must be maximal over the t-heavy edges. This t is at

least $w_e/2$, so if $e \notin M$, then either it shares an endpoint with some edge of weight at least $w_e/2$ or its own weight is $\leq \epsilon$. We then have

$$\begin{aligned} w(M') &= w(M \cap M') + \sum_{e \in M' \setminus M} w_e \\ &\leq w(M \cap M') + \sum_{e \in M' \setminus M} \max(2\delta_e, \epsilon) \\ &\leq w(M \cap M') + 2 \sum_{e \in M' \setminus M} \delta_e + \epsilon N \end{aligned}$$

We claim that $\sum_{e \in M' \setminus M} \leq 2 \cdot w(M \setminus M')$. This is because each edge in $M \setminus M$ shares an endpoint with at most 2 edges of $M' \setminus M$, otherwise M' would not be a matching. Therefore,

$$\begin{aligned} w(M') &\leq w(M \cap M') + 4 \sum_{e \in M \setminus M'} w_e + \epsilon N \\ &\leq 4 \cdot w(M) + \epsilon N \end{aligned}$$

By Lemma 28, $w(M') = N \cdot \mathrm{dist}_1(f, \mathrm{mono})$; therefore $w(M) \geq N(\frac{1}{4}\mathrm{dist}_1(f, \mathrm{mono}) - \epsilon)$ as desired.

We now bound the failure probability. When called with a random seed of length $\mathrm{poly}(\log N, \log\log(1/\epsilon))$ the algorithm of [Gha22] can be made to succeed with probability $1 - (N^{-10}/\log(4/\epsilon))$. We use the random seed on at most $\log(4/\epsilon)$ different graphs, so by union bound, with probability $1 - N^{-10}$ all the calls succeed. By the same argument as in the proof of Theorem 33, we may assume that $\log(1/\epsilon) \leq N$, and so the randomness complexity is $\mathrm{poly}(\Delta, \log N)$. □

5.6.8 Proof of Claim 11.

Let us first recall the statement of the claim:

Claim 11. *For any positive integers d and s, real $\epsilon, \delta \in (0,1)$, and any function $f : \{\pm 1\}^d \to [-1,1]$, let T be a collection of at least $d^{5s} \cdot \frac{100}{\epsilon^2} \ln \frac{1}{\epsilon} \ln \frac{1}{\delta}$ i.i.d. uniformly random elements of $\{\pm 1\}^d$. Then, with probability at least $1 - \delta$*

$$\max_{\substack{\text{degree-}s\text{ polynomial } P \\ \text{with } \|P\|_2 \leq 1}} \left| \|f - P\|_1 - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim T} [|f(\boldsymbol{x}) - P(\boldsymbol{x})|] \right| \leq \epsilon,$$

First we bound the probability that the condition above holds for one specific P with $\|P\|_2 \leq 1$. The condition $\|P\|_2 \leq 1$ implies that $\max_{\boldsymbol{x} \in \{\pm 1\}^d} |P(\boldsymbol{x})| \leq d^s$. This implies, via the Hoeffding bound, that

$$\Pr_{\text{choice of } T} \left[\left| \|f - P\|_1 - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim T} [|f(\boldsymbol{x}) - P(\boldsymbol{x})|] \right| > \frac{\epsilon}{4} \right] \leq \exp\left(-\frac{\epsilon^2}{32} \frac{|T|}{d^{2s}} \right).$$

We now move on to bounding the maximum over all degree-d polynomials P over $\{\pm 1\}^d$ with $\|P\|_2 \leq 1$. We will need a collection $\mathcal{C}$ of degree d polynomials over $\{\pm 1\}^d$, such that $|\mathcal{C}| \leq \exp\left(d^d \ln \frac{8d^d}{\epsilon}\right)$ so for every degree d polynomial P with $\|P\|_2 \leq 1$ there is some element $P_{\text{closest}} \in \mathcal{C}$ for which it is the case that

$$\max_{\boldsymbol{x} \in \{\pm 1\}^d} |P(\boldsymbol{x}) - P_{\text{closest}}(\boldsymbol{x})| \leq \frac{\epsilon}{4}.$$

Also, the L_2 norm of every element in $\mathcal{C}$ is at most 1. Such a set can be constructed by putting into $\mathcal{C}$ all polynomials of the form $\sum_{\substack{S \subset [d] \\ |S| \leq d}} c_S \left(\chi_S(x)\right)$ with the coefficients c_S taking values in $[-1, +1]$ rounded to the nearest multiple of $\frac{\epsilon}{8n^s}$, while discarding the polynomials whose L_2 norm is larger than 1. This way, since $\chi_S(x) \in \{\pm 1\}$, when we round the coefficients of P to a multiple of $\frac{\epsilon}{8d^s}$ the value at any $\boldsymbol{x} \in \{\pm 1\}^d$ cannot change by more than $\frac{\epsilon}{4}$, as there are at most d^s contributing monomials [17]. The total number of such polynomials is at most $\left(\frac{8d^s}{\epsilon}\right)^{d^s} = e^{d^s \ln \frac{8n^s}{\epsilon}}$.

Now, by taking a union bound on all elements of $\mathcal{C}$ we get

$$\Pr_{\text{choice of } T}\left[\max_{P \in \mathcal{C}} \left| \|f - P\|_1 - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] \right| \leq \frac{\epsilon}{2}\right] \geq 1 - \exp\left(-\frac{\epsilon^2}{32}\frac{|T|}{d^{2s}} + d^s \ln \frac{8d^s}{\epsilon}\right)$$

Finally, if the above holds, by choosing a polynomial P_{closest} from $\mathcal{C}$ to minimize $\max_{\boldsymbol{x} \in \{\pm 1\}^d} |P(\boldsymbol{x}) - P_{\text{closest}}(\boldsymbol{x})|$ we get that

$$\Pr_{\text{choice of } T}\left[\max_{\substack{\text{degree-}s \text{ polynomial } P \text{ over } \{\pm 1\}^d \\ \text{with } \|P\|_2 \leq 1}} \left| \|f - P\|_1 - \mathbb{E}_{\boldsymbol{x} \sim T}\left[|f(\boldsymbol{x}) - P(\boldsymbol{x})|\right] \right| \leq \epsilon\right] \geq$$

$$1 - \exp\left(-\frac{\epsilon^2}{8}\frac{|T|}{d^{2s}} + d^s \ln \frac{4d^s}{\epsilon}\right).$$

Substituting $|T| \geq d^{5s}\frac{100}{\epsilon^2} \ln \frac{1}{\epsilon} \ln \frac{1}{\delta}$ we see that the above expression is at least $1 - \delta$.

[17] To have $\|P_{\text{closest}}(\boldsymbol{x})\|_2 \leq \|P\|_2 \leq 1$ we should round to the closest multiple of $\frac{\epsilon}{8d^s}$ that is smaller in the absolute value of the coefficient being rounded

Bibliography

[AAK+07] Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k-wise and almost k-wise independence. *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 496–505, 2007.

[ABHU15] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, 40:167–190, 2015.

[ABHZ16] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, 49:152–192, 2016.

[ABL14] Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 449–458, 2014.

[ACSL07] Nir Ailon, Bernard Chazelle, C. Seshadhri, and Ding Liu. Estimating the distance to a monotone function. *Random Structures & Algorithms*, 31(3):371–383, 2007.

[ACSL08] Nir Ailon, Bernard Chazelle, C. Seshadhri, and Ding Liu. Property-Preserving Data Reconstruction. *Algorithmica*, 51(2):160–182, 2008.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Inf. Process. Lett.*, 88(3):107–110, 2003.

[AJMR14] Pranjal Awasthi, Madhav Jha, Marco Molinaro, and Sofya Raskhodnikova. Limitations of local filters of Lipschitz and monotone functions. *ACM Transactions on Computation Theory*, 7(1), December 2014. Publisher: Association for Computing Machinery (ACM).

[AL21] Rubi Arviv and Reut Levi. Improved LCAs for constructing spanners. *CoRR*, abs/2105.04847, 2021.

[AM91] William Aiello and Milena Mihail. Learning the Fourier spectrum of probabilistic lists and trees. *Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms*, pages 291–299, March 1991.

[AM06] Kazuyuki Amano and Akira Maruoka. On learning monotone Boolean functions under the uniform distribution. *Theor. Comput. Sci.*, 350(1):3–12, 2006.

[Ang88] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, apr 1988.

[ARVX12a] Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient local computation algorithms. *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1132–1139, 2012.

[ARVX12b] Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient Local Computation Algorithms. *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1132–1139, January 2012.

[Baz09] Louay MJ Bazzi. Polylogarithmic independence can fool dnf formulas. *SIAM Journal on Computing*, 38(6):2220–2272, 2009.

[BB21] Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. *SIAM J. Comput.*, 50(3), 2021.

[BBL98] Avrim Blum, Carl Burch, and John Langford. On Learning Monotone Boolean Functions. *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 408–415, 1998.

[BBL06] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Proceedings of the 23rd international conference on Machine learning*, pages 65–72, 2006.

[BCK+07] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. *Advances in neural information processing systems*, 20, 2007.

[BCO+15] Eric Blais, Clément L. Canonne, Igor C. Oliveira, Rocco A. Servedio, and Li-Yang Tan. Learning circuits with few negations. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, 40:512–527, 2015.

[BCS18] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $o(d) \cdot \text{polylog } n$ monotonicity tester for boolean functions over the hypergrid $[n]^d$. *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2133–2151, 2018.

[BCS20] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. Domain reduction for monotonicity testing: A $o(d)$ tester for boolean functions in d-dimensions. *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1975–1994, 2020.

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[BDBCP06] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[BDU12] Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. *International Conference on Algorithmic Learning Theory*, 2012.

[BF86] Imre Bárány and Zoltán Füredi. Computing the volume is difficult. *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 442–447, 1986.

[BFF+01] Tugkan Batu, Lance Fortnow, Eldar Fischer, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 442–451, 2001.

[BFR+00] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 259–269, 2000.

[BGJ+10] Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. *Approximation, Randomization, and Combinatorial Optimization*, pages 448–461, 2010.

[BGJ+12] Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-Closure Spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.

[BGR21] Sebastian Brandt, Christoph Grunau, and Václav Rozhon. The randomized local computation complexity of the Lovász local lemma. *CoRR*, abs/2103.16251, 2021.

[BH21] Maria-Florina Balcan and Nika Haghtalab. Noise in classification. *Beyond the Worst-Case Analysis of Algorithms*, page 361, 2021.

[BHV10] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine learning*, 80:111–139, 2010.

[Bir01] Lucien Birgé. An Alternative Point of View on Lepski's Method. *Lecture Notes-Monograph Series*, 36:113–133, 2001. Publisher: Institute of Mathematical Statistics.

[BK21] Ainesh Bakshi and Pravesh K Kothari. List-decodable subspace recovery: Dimension independent error in polynomial time. *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1279–1297, 2021.

[BLMT22] Guy Blanc, Jane Lange, Ali Malik, and Li-Yang Tan. On the power of adaptivity in statistical adversaries. *Conference on Learning Theory*, pages 5030–5061, 2022.

[BLQT22] Guy Blanc, Jane Lange, Mingda Qiao, and Li-Yang Tan. Properly learning decision trees in almost polynomial time. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 920–929, 2022.

[BM97] Lucien Birgé and Pascal Massart. From model selection to adaptive estimation. *Festschrift for lucien le cam*, pages 55–87, 1997.

[BNNR11] Khanh Do Ba, Huy L. Nguyen, Huy N. Nguyen, and Ronitt Rubinfeld. Sublinear time algorithms for earth mover's distance. *Theory Comput. Syst.*, 48(2):428–442, 2011.

[BOW08] E. Blais, R. O'Donnell, and K. Wimmer. Polynomial regression under arbitrary product distributions. *Machine Learning*, 2008.

[BRY14a] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 164–173, 2014.

[BRY14b] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 309–320, 2014.

[BT96] Nader H Bshouty and Christino Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM (JACM)*, 43(4):747–770, 1996. Publisher: ACM New York, NY, USA.

[BZ17] Maria-Florina F Balcan and Hongyang Zhang. Sample and computationally efficient learning algorithms under s-concave distributions. *Advances in Neural Information Processing Systems*, 30, 2017.

[CAL94] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

[Can22] Clément Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends® in Communications and Information Theory*, 19(6):1032–1198, 2022.

[CDST15] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean Function Monotonicity Testing Requires (Almost) $n^{1/2}$ Non-adaptive Queries. *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, June 2015.

[CFG+19] Yi-Jun Chang, Manuela Fischer, Mohsen Ghaffari, Jara Uitto, and Yufan Zheng. The complexity of (Δ+1) coloring in congested clique, massively parallel computation, and centralized local computation. *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 471–480, 2019.

[CGG+17] Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity. *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, 67:29:1–29:21, 2017.

[CGR13] Andrea Campagna, Alan Guo, and Ronitt Rubinfeld. Local reconstructors and tolerant testers for connectivity and diameter. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, 8096:411–424, 2013.

[CKKL12] Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K. Lee. Submodular Functions are Noise Stable. *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1586–1592, January 2012.

[CM13] T Tony Cai and Zongming Ma. Optimal hypothesis testing for high dimensional covariance matrices. *Bernoulli*, 19(5B):2359–2388, 2013.

[CS13a] Deeparnab Chakrabarty and C. Seshadhri. A o(n) monotonicity tester for boolean functions over the hypercube. *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, June 2013.

[CS13b] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 419–428, 2013.

[CS13c] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, 8096:425–435, 2013.

[CS19] Deeparnab Chakrabarty and C. Seshadhri. Adaptive boolean monotonicity testing in total influence time. *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, 124:20:1–20:7, 2019.

[CST14] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New Algorithms and Lower Bounds for Monotonicity Testing. *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, October 2014. ISSN: 0272-5428.

[CWX17] Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond Talagrand functions: new lower bounds for testing monotonicity and unateness. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, June 2017.

[Dan15] Amit Daniely. A PTAS for agnostically learning halfspaces. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, 40:484–502, 2015.

[Dan16] Amit Daniely. Complexity theoretic limitations on learning halfspaces. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 105–117, 2016.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM J. Comput.*, 39(8):3441–3462, 2010.

[DGK+21] Ilias Diakonikolas, Themis Gouleakis, Daniel M. Kane, John Peebles, and Eric Price. Optimal testing of discrete distributions with high probability. *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 542–555, 2021.

[DGL+99] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved Testing Algorithms for Monotonicity. *RANDOM-APPROX'99, Berkeley, CA, USA, August 8-11, 1999, Proceedings*, 1671:97–108, 1999.

[DGPP16] Ilias Diakonikolas, Themis Gouleakis, John Peebles, and Eric Price. Collision-based testers are optimal for uniformity and closeness. *Electron. Colloquium Comput. Complex.*, page 178, 2016.

[DGT19] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. *Advances in Neural Information Processing Systems*, 32, 2019.

[DHK+10] Ilias Diakonikolas, Prahladh Harsha, Adam Klivans, Raghu Meka, Prasad Raghavendra, Rocco A. Servedio, and Li-Yang Tan. Bounding the average sensitivity and noise sensitivity of polynomial threshold functions. *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, page 533, 2010.

[DK22] Ilias Diakonikolas and Daniel Kane. Near-optimal statistical query hardness of learning halfspaces with massart noise. *Conference on Learning Theory*, pages 4258–4282, 2022.

[DKK+21] Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Agnostic Proper Learning of Halfspaces under Gaussian Marginals. *Proceedings of Thirty Fourth Conference on Learning Theory*, pages 1522–1551, July 2021. ISSN: 2640-3498.

[DKK+22] Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning general halfspaces with general massart noise under the gaussian distribution. *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 874–885, 2022.

[DKK+23] Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Sihan Liu, and Nikos Zarifis. Efficient testable learning of halfspaces with adversarial label noise. *arXiv preprint arXiv:2303.05485*, 2023.

[DKMR22] Ilias Diakonikolas, Daniel Kane, Pasin Manurangsi, and Lisheng Ren. Cryptographic hardness of learning halfspaces with massart noise. *Advances in Neural Information Processing Systems*, 2022.

[DKPZ21] Ilias Diakonikolas, Daniel M. Kane, Thanasis Pittas, and Nikos Zarifis. The optimality of polynomial regression for agnostic learning under gaussian marginals in the SQ model. *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, 134:1552–1584, 2021.

[DKR23] Ilias Diakonikolas, Daniel M Kane, and Lisheng Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals. *arXiv preprint arXiv:2302.06512*, 2023.

[DKS18] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1061–1073, 2018.

[DKT21] Ilias Diakonikolas, Daniel Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. *Advances in Neural Information Processing Systems*, 34:7732–7744, 2021.

[DKTZ20a] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. *Conference on Learning Theory*, pages 1486–1513, 2020.

[DKTZ20b] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Nonconvex SGD learns halfspaces with adversarial label noise. *Advances in Neural Information Processing Systems*, 33:18540–18549, 2020.

[DKTZ22] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning general halfspaces with adversarial label noise via online gradient descent. *International Conference on Machine Learning*, pages 5118–5141, 2022.

[DKZ20] Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-optimal SQ lower bounds for agnostically learning halfspaces and relus under gaussian marginals. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[DLLP10] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.

[DSFT+14] Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. Proceedings, pages 498–511. Society for Industrial and Applied Mathematics, December 2014.

[DTK22] Ilias Diakonikolas, Christos Tzamos, and Daniel M Kane. A strongly polynomial algorithm for approximate forster transforms and its application to halfspace learning. *arXiv preprint arXiv:2212.03008*, 2022.

[Ele86] György Elekes. A geometric inequality and the complexity of computing volume. *Discret. Comput. Geom.*, 1:289–292, 1986.

[ELMR21] Guy Even, Reut Levi, Moti Medina, and Adi Rosén. Sublinear Random Access Generators for Preferential Attachment Graphs. *ACM Trans. Algorithms*, 17(4):28:1–28:26, 2021.

[EMR14] Guy Even, Moti Medina, and Dana Ron. Best of Two Local Models: Local Centralized and Local Distributed Algorithms. *CoRR*, abs/1402.3796, 2014.

[EYW12] Ran El-Yaniv and Yair Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13(2), 2012.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 563–574, 2006.

[FJS91] Merrick L. Furst, Jeffrey C. Jackson, and Sean W. Smith. Improved learning of AC^0 functions. *Proceedings of the fourth annual workshop on Computational learning theory*, pages 317–325, August 1991.

[FK15] Vitaly Feldman and Pravesh Kothari. Agnostic learning of disjunctions on symmetric distributions. *The Journal of Machine Learning Research*, 16(1):3455–3467, January 2015.

[FKP+19] Noah Fleming, Pravesh Kothari, Toniann Pitassi, et al. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[FKV17] Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. Tight Bounds on ℓ_1 Approximation and Learning of Self-Bounding Functions. *International Conference on Algorithmic Learning Theory*, pages 540–559, October 2017. ISSN: 2640-3498.

[FV15] V. Feldman and J. Vondrák. Tight Bounds on Low-Degree Spectral Concentration of Submodular and XOS Functions. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 923–942, October 2015. ISSN: 0272-5428.

[GGK20] Surbhi Goel, Aravind Gollakota, and Adam R. Klivans. Statistical-query lower bounds via functional gradients. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[GGL+00] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing Monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property Testing and its Connection to Learning and Approximation. *J. ACM*, 45(4):653–750, 1998.

[Gha15] Mohsen Ghaffari. An Improved Distributed Algorithm for Maximal Independent Set. *arXiv:1506.05093 [cs]*, July 2015. arXiv: 1506.05093.

[Gha22] Mohsen Ghaffari. Local computation of maximal independent set. *2022 IEEE 62nd Annual Symposium on Foundations of Computer Science*, 2022.

[GHL+15] Mika Göös, Juho Hirvonen, Reut Levi, Moti Medina, and Jukka Suomela. Non-local probes do not help with graph problems. *CoRR*, abs/1512.05411, 2015.

[GKK23] Aravind Gollakota, Adam R. Klivans, and Pravesh K. Kothari. A moment-matching approach to testable learning and a new characterization of rademacher complexity. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1657–1670, 2023.

[GKKM20] Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems*, 33:15859–15870, 2020.

[GKSV23] Aravind Gollakota, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Tester-learners for halfspaces: Universal algorithms. *37th Conference on Neural Information Processing Systems (NeurIPS 2023).*, 2023.

[GKSV24] Aravind Gollakota, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. An efficient tester-learner for halfspaces. *International Conference on Learning Representations (to appear)*, 2024.

[GOWZ10] Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling functions of halfspaces under product distributions. *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 223–234, 2010.

[GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electron. Colloquium Comput. Complex.*, (20), 2000.

[GR06] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 543–552, 2006.

[GR21] Jan Grebík and Václav Rozhon. Classification of local problems on paths from the perspective of descriptive combinatorics. *CoRR*, abs/2103.14112, 2021.

[GRSY20] Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. *Electron. Colloquium Comput. Complex.*, page 58, 2020.

[GS10] Parikshit Gopalan and Rocco A. Servedio. Learning and Lower Bounds for AC0 with Threshold Gates. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 588–601, 2010.

[GU19] Mohsen Ghaffari and Jara Uitto. Sparsifying Distributed Algorithms with Ramifications in Massively Parallel Computation and Centralized Local Computation. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1636–1653, 2019.

[Han07] Steve Hanneke. A bound on the label complexity of agnostic active learning. *Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007.

[Han09] Steve Hanneke. *Theoretical foundations of active learning*. Carnegie Mellon University, 2009.

[Han11] Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, pages 333–361, 2011.

[Han14] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[HJLT96] Thomas Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

[HKM10] Prahladh Harsha, Adam Klivans, and Raghu Meka. An invariance principle for polytopes. *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 543–552, June 2010.

[HL13] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

[JLSW11] Jeffrey C. Jackson, Homin K. Lee, Rocco A. Servedio, and Andrew Wan. Learning random monotone DNF. *Discret. Appl. Math.*, 159(5):259–271, 2011.

[Kan10] D. M. Kane. The Gaussian Surface Area and Noise Sensitivity of Degree-d Polynomial Threshold Functions. *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 205–210, June 2010. ISSN: 1093-0159.

[Kha80] Leonid G Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[KK21] Adam Tauman Kalai and Varun Kanade. Efficient learning with arbitrary covariate shift. *Algorithmic Learning Theory*, pages 850–864, 2021.

[KKK19a] Sushrut Karmalkar, Adam Klivans, and Pravesh Kothari. List-decodable linear regression. *Advances in neural information processing systems*, 32, 2019.

[KKK19b] Sushrut Karmalkar, Adam R. Klivans, and Pravesh Kothari. List-decodable linear regression. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7423–7432, 2019.

[KKM12] Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012.

[KKMS08] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008.

[KLS95] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13:541–559, 1995.

[KLS09] Adam R. Klivans, Philip M. Long, and Rocco A. Servedio. Learning halfspaces with malicious noise. *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, 5555:609–621, 2009.

[KMS15] Subhash Khot, Dor Minzer, and Muli Safra. On Monotonicity Testing and Boolean Isoperimetric Type Theorems. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, October 2015. ISSN: 0272-5428.

[KOS02] A. R. Klivans, R. O'Donnell, and R. A. Servedio. Learning intersections and thresholds of halfspaces. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 177–186, November 2002. ISSN: 0272-5428.

[KOS08] Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning geometric concepts via gaussian surface area. *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 541–550, 2008.

[KS88] Wieslaw Krakowiak and Jerzy Szulga. Hypercontraction principle and random multilinear forms. *Probability Theory and Related Fields*, 77(3):325–342, 1988.

[KS17] Pravesh K Kothari and Jacob Steinhardt. Better agnostic clustering via relaxed tensor norms. *arXiv preprint arXiv:1711.07465*, 2017.

[KSS94a] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. *Mach. Learn.*, 17(2-3):115–141, 1994.

[KSS94b] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994.

[KSV23] Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Testable learning with distribution shift. *Submitted to ArXiv*, 2023.

[KV89] Michael J. Kearns and Leslie G. Valiant. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 433–444, 1989.

[Las01] Jean B Lasserre. New positive semidefinite relaxations for nonconvex quadratic programs. *Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873–1950)*, pages 319–331, 2001.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

[LRR20] Reut Levi, Dana Ron, and Ronitt Rubinfeld. Local Algorithms for Sparse Spanning Graphs. *Algorithmica*, 82(4):747–786, 2020.

[LRV22] Jane Lange, Ronitt Rubinfeld, and Arsen Vasilyan. Properly learning monotone functions via local correction. *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, pages 75–86, 2022.

[LRY17] Reut Levi, Ronitt Rubinfeld, and Anak Yodpinyanee. Local Computation Algorithms for Graphs of Non-constant Degrees. *Algorithmica*, 77(4):971–994, 2017.

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065, 2015.

[LV07] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

[LV18] Yin Tat Lee and Santosh S Vempala. The Kannan-Lovász-Simonovits conjecture. *arXiv preprint arXiv:1807.03465*, 2018.

[LV23] Jane Lange and Arsen Vasilyan. Agnostic proper learning of monotone functions: beyond the black-box correction barrier. *in 64rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023.* ***Invited to special issue***, 2023.

[Man92] Yishay Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Proceedings of the fifth annual workshop on Computational learning theory*, pages 53–61, July 1992.

[MMR09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *Proceedings of The 22nd Annual Conference on Learning Theory (COLT 2009)*, 2009.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Nes00] Yurii Nesterov. Squared functional systems and optimization problems. *High performance optimization*, pages 405–440, 2000.

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[OS03] Ryan O'Donnell and Rocco A Servedio. New degree bounds for polynomial threshold functions. *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 325–334, 2003.

[OS06] R. O'Donnell and R. A. Servedio. Learning monotone decision trees in polynomial time. *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 13 pp.–225, July 2006. ISSN: 1093-0159.

[OW09] Ryan O'Donnell and Karl Wimmer. KKL, Kruskal-Katona, and Monotone Nets. *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 725–734, 2009.

[OZ18] Ryan O'Donnell and Yu Zhao. On closeness to k-wise uniformity. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, 116:54:1–54:19, 2018.

[Pan08] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inf. Theory*, 54(10):4750–4755, 2008.

[Par00] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[PRR04] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Electron. Colloquium Comput. Complex.*, 2004.

[PRVY19] Merav Parter, Ronitt Rubinfeld, Ali Vakilian, and Anak Yodpinyanee. Local computation algorithms for spanners. *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, 124:58:1–58:21, 2019.

[PRW22] Ramesh Krishnan S Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of Boolean functions. *Random Structures & Algorithms*, 60(2):233–260, 2022. Publisher: Wiley Online Library.

[Riv87] Ronald L. Rivest. Learning decision lists. *Mach. Learn.*, 2(3):229–246, 1987.

[RMH+20] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

[RTVX11a] Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. *ICS*, 2011.

[RTVX11b] Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast Local Computation Algorithms. *ICS*, 2011.

[RV16] Omer Reingold and Shai Vardi. New techniques and tighter bounds for local computation algorithms. *J. Comput. Syst. Sci.*, 82(7):1180–1200, 2016.

[RV23] Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1643–1656, 2023.

[RY20] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180, 2020.

[Sho87] N.Z. Shor. Quadratic optimization problems. *Izv. Akad. Nauk SSSR Tekhn. Kibernet.*, 1987(1):128–139, 222, 1987.

[SS10a] Michael Saks and C. Seshadhri. Local monotonicity reconstruction. *SIAM J. Comput.*, 39:2897–2926, 01 2010.

[SS10b] Michael Saks and C. Seshadhri. Local Monotonicity Reconstruction. *SIAM J. Comput.*, 39:2897–2926, January 2010.

[SW14] Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: a review. *Statistics surveys*, 8:45, 2014.

[TCK+22] Niels K Ternov, Anders N Christensen, Peter JT Kampen, Gustav Als, Tine Vestergaard, Lars Konge, Martin Tolsgaard, Lisbet R Hölmich, Pascale Guitera, Annette H Chakera, et al. Generalizability and usefulness of artificial intelligence for skin cancer diagnostics: An algorithm validation study. *JEADV Clinical Practice*, 1(4):344–354, 2022.

[Tre19] Lloyd N Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[Wim10] K. Wimmer. Agnostically Learning under Permutation Invariant Distributions. *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 113–122, October 2010. ISSN: 0272-5428.

[WOD+21] Andrew Wong, Erkin Otles, John P Donnelly, Andrew Krumm, Jeffrey McCullough, Olivia DeTroyer-Cooley, Justin Pestrue, Marie Phillips, Judy Konye, Carleen Penoza, et al. External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients. *JAMA Internal Medicine*, 181(8):1065–1070, 2021.

[Wol07] Paweł Wolff. Hypercontractivity of simple random variables. *Studia Mathematica*, 3(180):219–236, 2007.

[YBC13] Liu Yang, Avrim Blum, and Jaime Carbonell. Learnability of DNF with representation-specific queries. *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, page 37–46, 2013.

[YZ17] Songbai Yan and Chicheng Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1056–1066, 2017.

[ZBL+18] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

[Zha18] Chicheng Zhang. Efficient active learning of sparse halfspaces. *Conference on Learning Theory*, pages 1856–1880, 2018.

[ZL21] Chicheng Zhang and Yinan Li. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. *Conference on Learning Theory*, pages 4526–4527, 2021.

[ZSA20] Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. *Advances in Neural Information Processing Systems*, 33:7184–7197, 2020.