The Dissertation Committee for Aravind Gollakota
certifies that this is the approved version of the following dissertation:

New Computational and Statistical Characterizations of Neural Network Learning

Committee:

Adam R. Klivans, Supervisor

Pravesh K. Kothari

Qiang Liu

Eric Price

New Computational and Statistical Characterizations of Neural Network Learning

by

Aravind Gollakota

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2023

Acknowledgments

I owe a great deal to many wonderful people who made this PhD possible. Thanks go above all to my advisor, Adam Klivans, for welcoming me in and giving me the space to discover my own interests, for his easygoing approach and sense of humor in both research and life, for his excellent taste in problems, and for believing in me even through the lows and the lulls.

Collaboration is one of the great pleasures of research, and I feel truly fortunate and grateful to have been able to work with a stellar list of colleagues during my time in grad school: Sitan Chen, Yuval Dagan, Giannis Daras, Alex Dimakis, Surbhi Goel, Parikshit Gopalan, Zhihan Jin, Sushrut Karmalkar, Adam Klivans, Pravesh Kothari, Daniel Liang, Raghu Meka, Kulin Shah, Konstantinos Stavropoulos, and Arsen Vasilyan. Surbhi and Sushi showed me the ropes when I came in; and Kostas and Kulin came in not needing to be shown the ropes at all. I'm also very grateful to those who mentored me through my tentative first steps in research: Julia Chuzhoy, Karola Mészáros, István Miklós, David Steurer, and Madhur Tulsiani.

Austin has been a fantastic home for the last six years, and I will cherish the many memories from my time spent here with a fabulous group of friends, among them Akanksha, Akshay, Ananya, Daniel, Eric, Josh, Justin, Kevin, Kostas, Kulin, Matt, Meghna, Raghava, Ridwan, Sepideh, Shivam, Surbhi, and, of course, Sushi. To my friends from Cornell (Abhijit, Akshay, Athith, Jun Wei, Neha, Nivedita, Sandeep, Soham, Somrita, Sush, and many more), and from back home in India (Aditi, Athyuttam, Mihir, Prateek, Shubham, and many more): thank you guys for being there.

Heartfelt thanks go also to my extended family both near and far for their love and invaluable support, and especially to my uncles Rama and Ravi for helping to make the journey to these shores possible at all.

Finally, to my parents, for their unceasing love, encouragement, faith, and wisdom, and for everything else besides: thank you so much.

Abstract

New Computational and Statistical Characterizations of Neural Network Learning

Aravind Gollakota, PhD
The University of Texas at Austin, 2023

SUPERVISOR: Adam R. Klivans

A foundational goal of machine learning theory is to characterize the inherent computational and statistical complexity of some of the most basic tasks in machine learning. In this thesis, we present new results concerning two such tasks in neural network learning and beyond.

First, we study the question of when efficient algorithms can achieve high test accuracy on labeled data known to be consistent with a simple neural network. We present a set of results establishing the surprising computational intractability of this problem even in the benign setting where the inputs are drawn from a Gaussian, and the labels are perfectly consistent with a simple two-hidden-layer or even one-hidden-layer neural network. These hardness results illuminate what types of problem assumptions are necessary for efficient algorithms for this problem to be possible at all.

Next, we investigate the problem of testing whether a learning algorithm has fit the data as well as its guarantee claims. This is a serious issue for agnostic supervised learning (i.e. supervised learning with no assumptions on the labels), where most efficient algorithms make simplifying distributional assumptions such as Gaussianity. But such assumptions can be hard to verify, meaning it can be hard to check whether

the learner has actually succeeded. The recent elegant model of testable learning addresses this issue by replacing such hard-to-verify distributional assumptions with efficiently testable ones. We present both a broad algorithmic framework as well as a full statistical characterization of this model.

Table of Contents

Chapter 1: Introduction

A foundational goal of machine learning theory is to characterize the complexity of some of the most basic tasks in machine learning. When can we achieve high test accuracy on data known to be consistent with a simple neural network or Boolean function? When can we be sure a learning algorithm has fit the data as well as possible? This thesis explores these and other related questions using tools from computational and statistical learning theory as well as complexity theory. The focus is on studying the inherent computational complexity of key learning problems. The goal is both to develop new algorithms and to show lower bounds that illuminate what types of problem assumptions are necessary for efficient algorithms to be possible at all.

1.1 When can we achieve high test accuracy on data known to be consistent with a simple neural network?

Consider for a moment the remarkable fact that given a large labeled training dataset of images, one can easily train a deep neural network to achieve extremely high accuracy at, say, detecting whether a fresh test image depicts a human face. What explains this performance? Certainly the fact that high test accuracy is achieved tells us that there *exists* a deep neural network nearly consistent with the labeled data; this might naturally lead us to study the approximation power of neural networks. Moreover, we have also managed to efficiently *nd* such a network; this raises the important question of optimization for neural networks. This thesis adopts the perspective of *computational learning theory* [Val84, Vap98]: even in the ideal case where the data is *exactly* consistent with a simple neural network — i.e., given labeled data (x, y) *2* R^d R drawn from a distribution *D* where $y = f\ (x)$ for some unknown but simple neural network f — when is it possible to efficiently learn *any* hypothesis (not necessarily a neural network) that achieves high test accuracy? This is one of the

most basic definitions in learning theory, also known as the Probably Approximately Correct (PAC) model.

An especially useful model for such problems is the Statistical Query (SQ) model [Kea98, BFJ+94, Rey20]. SQ learners are abstractions of many common learning algorithms (including gradient descent) that access the data exclusively through statistical queries of the form $\mathsf{E}_{(x,y)\ D}[\phi(x,y)]\quad\tau$ for some bounded query function ϕ and tolerance τ. In recent years, there has been a substantial body of work on showing lower bounds and statistical-computational tradeoffs for many supervised and unsupervised learning problems (see e.g. [DKS17, BB20, BBH+20] and references therein), and for such problems SQ lower bounds are viewed as strong evidence of the limits of computationally efficient learners.

In Chapter 2, we show that this problem is hard for a broad subclass of SQ learners known as correlational SQ (CSQ) learners (which include gradient descent on squared loss) even in the benign setting where the distribution over inputs is the standard Gaussian and f is just a one-hidden-layer network of the form $f(x) = \sum_{i=1}^{m} a_i\sigma(w_i\quad x)$. This result, however, leaves open the question of ruling out *all* SQ learners (as opposed to just the CSQ subclass), and prevailing wisdom (see e.g. [VW19a]) actually suggested that this might not even be possible for a real-valued function family in the absence of noise. In Chapter 3, we answer this question and show the first *general* SQ lower bounds for learning noise-free *two*-hidden-layer ReLU networks over the Gaussian. In fact, we show that the same hard family of two-hidden-layer networks is also hard for *any* efficient learner (not necessarily SQ) under a cryptographic assumption. We also show the first hardness result for learning constant-depth networks using *label queries.*

On the other end of the spectrum from the noise-free setting is the agnostic setting, where the input distribution is still Gaussian but the labels may be arbitrarily noisy. For this setting, we show in Chapter 4 that the problem becomes harder still: merely finding a hypothesis competitive with the best-fitting *single ReLU* is beyond

any efficient SQ algorithm. Our proof introduces a novel algorithm for lifting a learner for single neurons to a learner for one-hidden-layer networks using Frank–Wolfe functional gradient descent.

Taken together, these constitute the strongest lower bounds to date on learning shallow neural networks over the Gaussian. From an algorithm designer's standpoint, the value of these lower bounds is in showing what types of problem assumptions it is necessary to make for efficient algorithms to be possible at all.

1.2 What general characterizations of learning can we provide in the SQ model?

Moving beyond neural network learning, it is natural to wonder whether one can provide characterizations of the learnability of general Boolean concept classes, at least in the SQ model. In the distribution-free setting (in which the input distribution is unknown and arbitrary), the only known efficient learners for almost all rich classes are based on the so-called "polynomial method" [HS07]. This method relies on approximating Boolean functions by low-degree polynomials (in various senses depending on the precise setting), and then using simple learners for these polynomial representations. Is this the best we can do?

Remarkably, there is evidence suggesting that the answer is yes: there exist SQ lower bounds that essentially match the upper bounds given by the polynomial method. These lower bounds follow as consequences of various powerful results due to [Fel08, Fel12, She08b, She11], leveraging connections to other fields such as communication complexity, and hold primarily for the subclass of correlational SQ (CSQ) learners. While most known PAC algorithms fall into the SQ framework, it is worth pointing out that CSQ learners are only a subset of SQ learners, and there do exist results showing separations between CSQ and SQ learnability [Fel08]. Nevertheless, such CSQ lower bounds do constitute important evidence of hardness of learning in the general PAC model.

For such a basic result, alternative proofs are always useful, and in Chapter 5 we provide a new proof of the optimality of the polynomial method in the CSQ model. Our formal lower bound for noise-free/realizable learning is stated in terms of the so-called threshold degree, namely the minimum degree required for polynomials to approximate a given concept class in 0/1 loss (i.e. in sign), and for agnostic learning in terms of the approximate degree, namely the minimum degree required for polynomials to approximate a class pointwise. To our knowledge, the precise statements we give had not appeared in this form before, and help to further clarify the role of the polynomial method in computational learning theory. Moreover, our proofs are simple and largely self-contained.

1.3 When can we be sure a learning algorithm has fit the data as well as possible?

For many challenging problems in machine learning theory, simplifying assumptions are made in search of efficient algorithms. But an underappreciated fact about such assumptions is that if they cannot be efficiently verified, then for some problems it can be hard to verify *whether the learning algorithm has actually succeeded*. This is a significant issue for the usability as well as interpretability of these algorithms.

An important example is agnostic supervised learning, where the learner's goal is produce a hypothesis with error close to the optimal error $\mathsf{opt}(C, D)$ achievable by any function in a function class C over a labeled distribution D. To make this problem tractable, it is common to make distributional assumptions such as Gaussianity. But these assumptions can be hard or impossible to verify given only samples from D, and because we have no prior knowledge of $\mathsf{opt}(C, D)$, it is not clear how to verify the learner's error guarantee.

The recent elegant model of *testable learning* [RV23] addresses this issue by requiring that the learner succeed whenever an associated efficient tester accepts the

unknown distribution. Moreover, the tester is required to accept whenever the original distributional assumption that is being replaced is indeed satisfied.

In Chapter 6, we establish some foundational results for this model. We show a powerful algorithmic framework that yields testable learners matching the bounds of the optimal ordinary agnostic learners for many important classes. Our framework works for any function class that admits *sandwiching polynomials* with bounded coefficients, and may be viewed as an analog of the universal approach to agnostic learning based on L^1 polynomial regression of [KKMS08]. The associated tester is simple and relies entirely on checking low-degree empirical moments. On the sample complexity front, we show a nearly-tight characterization of the sample complexity of testable learning in terms of Rademacher complexity, a well-studied quantity in statistical learning theory [BBL03]. Intriguingly, this means uniform convergence is both necessary and sufficient for testable learning, yielding a fundamental separation from ordinary distribution-specific agnostic learning. This is the first natural model-based characterization of Rademacher complexity. This separation also arguably gives us a new lens on understanding generalization in modern overparameterized models [BMR21, Bel21].

Bibliographic note

This thesis is based primarily on the following published papers, appearing with the permission of the coauthors:

- Chapter 2 is based on joint work with Surbhi Goel, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. This work appeared in ICML 2020 [GGJ+20].
- Chapter 3 is based on joint work with Sitan Chen, Raghu Meka, and Adam Klivans. This work appeared in NeurIPS 2022 [CGMK22].
- Chapter 4 is based on joint work with Surbhi Goel and Adam Klivans. This work appeared in NeurIPS 2020 [GGK20].

- Chapter 6 is based on joint work with Pravesh Kothari and Adam Klivans. This work appeared in STOC 2023 [GKK23].

Chapter 2: Superpolynomial Lower Bounds for Learning One-Layer Neural Networks using Gradient Descent

2.1 Introduction

A major challenge in the theory of deep learning is to understand when gradient descent can efficiently learn simple families of neural networks. The associated optimization problem is nonconvex and well known to be computationally intractable in the worst case. For example, cyphertexts from public-key cryptosystems can be encoded into a training set labeled by simple neural networks [KS09], implying that the corresponding learning problem is as hard as breaking cryptographic primitives. These hardness results, however, rely on discrete representations and produce relatively unrealistic joint distributions.

2.1.1 Our Results

In this chapter we give the first superpolynomial lower bounds for learning neural networks using gradient descent in arguably the simplest possible setting: we assume the marginal distribution is a spherical Gaussian, the labels are noiseless and are exactly equal to the output of a one-layer neural network (a linear combination of say ReLU or sigmoid activations), and the goal is to output a classifier whose test error (measured by square-loss) is small. We prove—unconditionally—that gradient descent fails to produce a classifier with small square-loss if it is required to run in polynomial time in the dimension. Our lower bound depends only on the algorithm used (gradient descent) and not on the architecture of the underlying classifier. That is, our results imply that current popular heuristics such as running gradient descent on an overparameterized network (for example, working in the NTK regime [JHG18]) will require superpolynomial time to achieve small test error.

Statistical Queries. We prove our lower bounds in the now well-studied statistical query (SQ) model of [Kea98] that captures most learning algorithms used in practice. For a loss function ℓ and a hypothesis h_θ parameterized by θ, the true population loss with respect to joint distribution D on $X \quad Y$ is given by $\mathsf{E}_{(x,y)\ \ D}[\ell(h_\theta(x), y)]$, and the gradient with respect to θ is given by $\mathsf{E}_{(x,y)\ \ D}[\ell'(h_\theta(x), y)\, r_\theta h_\theta(x)]$. In the SQ model, we specify a query function $\phi(x, y)$ and receive an estimate of $j\mathsf{E}_{(x,y)\ \ D}[\phi(x, y)]j$ to within some tolerance parameter τ. An important special class of queries are correlational or inner-product queries, where the query function ϕ is defined only on X and we receive an estimate of $j\mathsf{E}_{(x,y)\ \ D}[\phi(x) \quad y]j$ within some tolerance τ. It is not difficult to see that (1) the gradient of a population loss can be approximated to within τ using statistical queries of tolerance τ and (2) for square-loss only inner-product queries are required.

Since the convergence analysis of gradient descent holds given sufficiently strong approximations of the gradient, lower bounds for learning in the SQ model [Kea98, BFJ+94, Szö09, Fel12, Fel17] directly imply unconditional lower bounds on the running time for gradient descent to achieve small error. We give the first super-polynomial lower bounds for learning one-layer networks with respect to any Gaussian distribution for any SQ algorithm that uses inner product queries:

Theorem 2.1.1 (informal)**.** *Let C be a class of real-valued concepts de ned by one-layer single-output neural networks with input dimension n and m hidden units (ReLU or sigmoid); i.e., functions of the form $f(x) = \sum_{i=1}^{m} a_i \sigma(w_i\ x)$. Then learning C under the standard Gaussian $N(0, I_n)$ in the SQ model with inner-product queries requires $n^{(\log m)}$ queries for any tolerance $\tau = n^{-(\log m)}$.*

In particular, this rules out any approach for learning one-layer neural networks in polynomial-time that performs gradient descent on any polynomial-size classifier with respect to square-loss or logistic loss. For classification, we obtain significantly stronger results and rule out general SQ algorithms that run in polynomial-time (e.g.,

gradient descent with respect to any polynomial-size classifier and any polynomial-time computable loss). In this setting, our labels are $f \quad 1g$ and correspond to the *softmax* of an unknown one-layer neural network. We prove the following:

Theorem 2.1.2 (informal)**.** *Let C be a class of real-valued concepts de ned by a one-layer neural network in n dimensions with m hidden units (ReLU or sigmoid) feeding into any odd, real-valued output node with range $[\ 1, 1]$. Let $D^{\emptyset}$ be a distribution on $\mathsf{R}^n \quad f \ 1g$ such that the marginal on R^n is the standard Gaussian $N(0, I_n)$, and $\mathsf{E}[Y jX] = c(X)$ for some $c \ 2 \ C$. For some $b, C > 0$ and $\epsilon = Cm^{\ b}$, outputting a classi er $f : \mathsf{R}^n \ ! \ f \ 1g$ with $\mathsf{P}_{(X,Y) \ D^{\emptyset}}[f(X) \ 6= Y] \quad 1/2 \quad \epsilon$ requires $n^{\ (\log m)}$ statistical queries of tolerance $n^{\ (\log m)}$.*

The above lower bound for classification rules out the commonly used approach of training a polynomial-size, real-valued neural network using gradient descent (with respect to any polynomial-time computable loss) and then taking the sign of the output of the resulting network.

2.1.2 Our techniques

At the core of all SQ lower bounds is the construction of a family of functions that are pairwise approximately orthogonal with respect to the underlying marginal distribution. Typically, these constructions embed 2^n parity functions over the discrete hypercube $f \quad 1, 1g^n$. Since parity functions are perfectly orthogonal, the resulting lower bound can be quite strong. Here we wish to give lower bounds for more natural families of distributions, namely Gaussians, and it is unclear how to embed parity.

Instead, we use an alternate construction. For activation functions $\phi, \psi : \mathsf{R} \ ! \ \mathsf{R}$, define

$$f_S(x) = \psi\left(\sum_{w2f \ \ 1,1g^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right)\right).$$

Enumerating over every S $[n]$ of size k gives a family of functions of size $n^{O(k)}$. Here x_S denotes the vector of x_i for i 2 S (typically we choose $k = \log m$ to produce a family of one-layer neural networks with m hidden units). Each of the $2^k = m$ inner weight vectors are all of unit norm, and all of the m outer weights have absolute value one. Note also that our construction uses activations with zero bias term.

We give a complete characterization of the class of nonlinear activations for which these functions are orthogonal. In particular, the family is orthogonal for any activation with a nonzero Hermite coefficient of degree k or higher.

Apart from showing orthogonality, we must also prove that functions in these classes are nontrivial (i.e., are not exponentially close to the constant zero function). This reduces to proving certain lower bounds on the norms of one-layer neural networks. The analysis requires tools from Hermite and complex analysis.

SQ Lower Bounds for Real-Valued Functions. Another major challenge is that our function family is real-valued as opposed to boolean. Given an orthogonal family of (deterministic) *boolean* functions, it is straightforward to apply known results and obtain general SQ lower bounds for learning with respect to 0/1 loss. For the case of real-valued functions, the situation is considerably more complicated. For example, the class of orthogonal Hermite polynomials on n variables of degree d has size $n^{O(d)}$, yet there *is* an SQ algorithm due to [APVZ14] that learns this class with respect to the Gaussian distribution in time $2^{O(d)}$. More recent work due to [ADHV19] shows that Hermite polynomials can be learned by an SQ algorithm in time polynomial in n and $\log d$.

As such, it is *impossible* to rule out general polynomial-time SQ algorithms for learning real-valued functions based solely on orthogonal function families. Fortunately, it is not difficult to see that the SQ reductions due to [Szö09] hold in the real-valued setting as long as the learning algorithm uses only *inner-product queries* (and the norms of the functions are sufficiently large). Since performing gradient

descent with respect to square-loss or logistic loss can be implemented using inner-product queries, we obtain our first set of desired results[1].

Still, we would like rule out *general* SQ algorithms for learning simple classes of neural networks. To that end, we consider the *classi cation* problem for one-layer neural networks and output labels after performing a *softmax* on a one-layer network. Concretely, consider a distribution on $\mathsf{R}^n \quad f \quad 1, 1g$ where $\mathsf{E}[YjX] = \sigma(c(X))$ for some $c \; 2 \; C$ and $\sigma : \mathsf{R} \; ! \; [\quad 1, 1]$ (for example, σ could be tanh). We describe two goals. The first is to estimate the conditional mean function, i.e., output a classifier h such that $\mathsf{E}[(h(x) \quad c(x))^2] \quad \epsilon$. The second is to directly minimize classification loss, i.e., output a boolean classifier h such that $\mathsf{P}_{X,Y \quad D}[h(X) \; \not= \; Y] \quad 1/2 \quad \epsilon$.

We give superpolynomial lower bounds for both of these problems in the general SQ model by making a new connection to *probabilistic concepts*, a learning model due to [KS94a]. Our key theorem gives a superpolynomial SQ lower bound for the problem of *distinguishing* probabilistic concepts induced by our one-layer neural networks from truly random labels. A final complication we overcome is that we must prove orthogonality and norm bounds on one-layer neural networks that have been composed with a nonlinear activation (e.g., tanh).

SGD and Gradient Descent Plus Noise. It is easy to see that our results also imply lower bounds for algorithms where the learner adds noise to the estimate of the gradient (e.g., Langevin dynamics). On the other hand, for technical reasons, it is known that SGD is *not* a statistical query algorithm (because it examines training points individually) and does not fall into our framework. That said, recent work by [AS20] shows that SGD is *universal* in the sense that it can encode all polynomial-time learners. This implies that proving unconditional lower bounds for SGD would give a proof that $\mathsf{P} \; \not= \; \mathsf{NP}$. Thus, we cannot hope to prove unconditional lower bounds on SGD (unless we can prove $\mathsf{P} \; \not= \; \mathsf{NP}$).

[1] The algorithms of [APVZ14] and [ADHV19] do not use inner-product queries.

2.1.3 Related Work

There is a large literature of results proving hardness results (or unconditional lower bounds in some cases) for learning various classes of neural networks [BR89, Vu98, KS09, LSSS14, GKKT17b].

The most relevant prior work is due to [SVWX17], who addressed learning one-layer neural networks under logconcave distributions using Lipschitz queries. Specifically, let n be the input dimension, and let m be the number of hidden s-Lipschitz sigmoid units. For $m = \tilde{O}(s^{p}\overline{n})$, they construct a family of neural networks such that any learner using λ-Lipschitz queries with tolerance greater than $\Omega(1/(s^2 n))$ needs at least $2^{(n)}/(\lambda s^2)$ queries.

Roughly speaking, their lower bounds hold for λ-Lipschitz queries due to the composition of their one-layer neural networks with a δ-function in order make the family more "boolean." Because of their restriction on the tolerance parameter, they cannot rule out gradient descent with large batch sizes. Further, the slope of the activations they require in their constructions scales inversely with the Lipschitz and tolerance parameters.

To contrast with [SVWX17], note that our lower bounds hold for any inverse-polynomial tolerance parameter (i.e., *will* hold for polynomially-large batch sizes), do not require a Lipschitz constraint on the queries, and use only standard 1-Lipschitz ReLU and/or sigmoid activations (with zero bias) for the construction of the hard family. Our lower bounds are typically quasipolynomial in the number of hidden units; improving this to an exponential lower bound is an interesting open question. Both of our models capture square-loss and logistic loss.

In terms of techniques, [SVWX17] build an orthogonal function family using univariate, periodic "wave" functions. Our construction takes a different approach, adding and subtracting activation functions with respect to overlapping "masks." Finally, aside from the (black-box) use of a theorem from complex analysis, our construction and analysis are considerably simpler than the proof in [SVWX17].

A follow-up work [VW19b] gave SQ lower bounds for learning classes of degree d orthogonal polynomials in n variables with respect to the uniform distribution on the unit sphere (as opposed to Gaussians) using inner product queries of bounded tolerance (roughly $1/n^d$). To obtain superpolynomial lower bounds, each function in the family requires superpolynomial description length (their polynomials also take on very small values, $1/n^d$, with high probability).

Shamir [Sha18a] (see also the related work of [SSSS17]) proves hardness results (and lower bounds) for learning neural networks using gradient descent with respect to square-loss. His results are separated into two categories: (1) hardness for learning "natural" target families (one layer ReLU networks) or (2) lower bounds for "natural" input distributions (Gaussians). We achieve lower bounds for learning problems with *both* natural target families and natural input distributions. Additionally, our lower bounds hold for any nonlinear activations (as opposed to just ReLUs) and for broader classes of algorithms (SQ).

Recent work due to [GKK19] gives hardness results for learning a ReLU with respect to Gaussian distributions. Their results require the learner to output a single ReLU as its output hypothesis and require the learner to succeed in the agnostic model of learning. [KK14b] prove hardness results for learning a threshold function with respect to Gaussian distributions, but they also require the learner to succeed in the agnostic model. Very recent work due to Daniely and Vardi [DV20a] gives hardness results for learning randomly chosen two-layer networks. The hard distributions in their case are not Gaussians, and they require a nonlinear clipping output activation.

Positive Results. Many recent works give algorithms for learning one-layer ReLU networks using gradient descent with respect to Gaussians under various assumptions [ZSJ+17a, ZPS17, BG17a, ZYWG19a] or use tensor methods [JSA15, GLM18a].

These results depend on the hidden weight vectors being sufficiently orthogonal, or the coefficients in the second layer being positive, or both. Our lower bounds explain why these types of assumptions are necessary.

Independent Work. Independently, Diakonikolas et al. [DKKZ20] have given stronger correlational SQ lower bounds for the same class of functions with respect to the Gaussian distribution. Their bounds are exponential in the number of hidden units while ours is quasipolynomial. We can plug in their result and obtain exponential general SQ lower bounds for the associated probabilistic concept using our framework.

2.2 Preliminaries

We use $[n]$ to denote the set $\{1, \ldots, n\}$, and $S \subseteq_k T$ to indicate that S is a k-element subset of T. We denote euclidean inner products between vectors u and v by $u \cdot v$. We denote the element-wise product of vectors u and v by $u \circ v$, that is, $u \circ v$ is the vector $(u_1 v_1, \ldots, u_n v_n)$.

Let X be an arbitrary domain, and let D be a distribution on X. Given two functions $f, g : X \to \mathbb{R}$, we define their L_2 inner product with respect to D to be $\langle f, g \rangle_D = \mathbb{E}_D[fg]$. The corresponding L_2 norm is given by $\|f\|_D = \sqrt{\langle f, f \rangle_D} = \sqrt{\mathbb{E}_D[f^2]}$.

A real-valued concept on $\mathbb{R}^n$ is a function $c : \mathbb{R}^n \to \mathbb{R}$. We denote the induced labeled distribution on $\mathbb{R}^n \times \mathbb{R}$, i.e. the distribution of $(x, c(x))$ for $x \in D$, by D_c. A probabilistic concept, or p-concept, on X is a concept that maps each point x to a random $\{-1, 1\}$-valued label in such a way that $\mathbb{E}[Y|X] = c(X)$ for a fixed function $c : \mathbb{R}^n \to [-1, 1]$, known as the conditional mean function. Given a distribution D on the domain, we abuse D_c to denote the induced labeled distribution on $X \times \{-1, 1\}$ such that the marginal distribution on $\mathbb{R}^n$ is D and $\mathbb{E}[Y|X] = c(X)$ (equivalently the label is $+1$ with probability $\frac{1+c(x)}{2}$ and -1 otherwise).

2.2.1 The SQ model

A statistical query is specified by a query function $h : \mathsf{R}^d \quad Y \;!\; \mathsf{R}$. The SQ model allows access to an SQ oracle that accepts a query h of specified tolerance τ, and responds with a value in $[\mathsf{E}_{x,y}[h(x,y)] \quad \tau, \mathsf{E}_{x,y}[h(x,y)] + \tau]$.[2] To disallow arbitrary scaling, we will require that for each y, the function $x \;?\; h(x,y)$ has norm at most 1. In the real-valued setting, a query h is called a correlational or inner product query if it is of the form $h(x,y) = g(x) \quad y$ for some function g, so that $\mathsf{E}_{D_c}[h] = \mathsf{E}_D[gc] = \hbar g, c\imath_D$. Here we assume $\|g\| \quad 1$ when stating lower bounds, again to disallow arbitrary scaling.

Gradient descent with respect to squared loss is captured by inner product queries, since the gradient is given by

$$\begin{aligned}\mathsf{E}_{x,y}[\,\nabla_\theta(h_\theta(x) \quad y)^2] &= \mathsf{E}_{x,y}[2(h_\theta(x) \quad y)\,\nabla_\theta h_\theta(x)] \\ &= 2\,\mathsf{E}_x[h_\theta(x)\,\nabla_\theta h_\theta(x)] \\ &\quad 2\,\mathsf{E}_{x,y}[y\,\nabla_\theta h_\theta(x)].\end{aligned}$$

Here the first term can be estimated directly using knowledge of the distribution, while the latter is a vector each of whose elements is an inner product query.

We now formally define the learning problems we consider.

Definition 2.2.1 (SQ learning of real-valued concepts using inner product queries)**.** Let C be a class of p-concepts over a domain X, and let D be a distribution on X. We say that a learner learns C with respect to D up to L_2 error ϵ using inner product quiers (equivalently squared loss ϵ^2) if, given only SQ oracle access to D_c for some unknown $c \;2\; \mathsf{C}$, and using only inner product queries, it is able to output $\tilde{c} : X \;!\; [\quad 1, 1]$ such that $\|c \quad \tilde{c}\|_D \quad \epsilon$.

[2] In the SQ literature, this is referred to as the STAT oracle. A variant called VSTAT is also sometimes used, known to be equivalent up to small polynomial factors [Fel17]. While it makes no substantive di erence to our superpolynomial lower bounds, our arguments can be extended to VSTAT as well.

For the classification setting, we consider two different notions of learning p-concepts. One is learning the target up to small L_2 error, to be thought of as a strong form of learning. The other, weaker form, is achieving a nontrivial inner product (i.e. unnormalized correlation) with the target. We prove lower bounds on both in order to capture different learning goals.

Definition 2.2.2 (SQ learning of p-concepts)**.** Let $\mathcal{C}$ be a class of p-concepts over a domain X, and let D be a distribution on X. We say that a learner learns $\mathcal{C}$ with respect to D up to L_2 error ϵ if, given only SQ oracle access to D_c for some unknown $c \in \mathcal{C}$, and using arbitrary queries, it is able to output $\tilde{c} : X \to [-1, 1]$ such that $\|c - \tilde{c}\|_D \leq \epsilon$. We say that a learner weakly learns $\mathcal{C}$ with respect to D with advantage ϵ if it is able to output $\tilde{c} : X \to [-1, 1]$ such that $|\langle \tilde{c}, c \rangle_D| \geq \epsilon$.

Note that the best achievable advantage is $\mathbf{E}_{x \sim D}[|c(x)|]$, achieved by $\tilde{c}(x) = \text{sign}(c(x))$. Note also that $\|c\|_D^2 \leq \mathbf{E}_D[|c|] \leq \|c\|_D$, and therefore a norm lower bound on functions in $\mathcal{C}$ implies an upper bound on the achievable advantage.

Remark 2.2.3 (Learning with L_2 error implies weak learning)*.* If the functions in our class satisfy a norm lower bound, say $\|c\|_D^2 \geq (1 + \alpha)\epsilon^2$, then a simple calculation shows that learning with L_2 error ϵ implies weak learning with advantage $\alpha\epsilon^2/2$.

Our definition of weak learning also captures the standard boolean sense of weak learning, in which the learner is required to output a boolean hypothesis with 0/1 loss bounded away from 1/2. Indeed, by an easy calculation, the 0/1 loss of a function $f : X \to \{-1, 1\}$ satisfies

$$\mathbf{P}_{(x,y) \sim D_c}[f(x) \neq y] = \frac{1}{2} - \frac{|\langle c, f \rangle_D|}{2}.$$

The difficulty of learning a concept class in the SQ model is captured by a parameter known as the statistical dimension of the class.

Definition 2.2.4 (Statistical dimension)**.** Let $\mathcal{C}$ be a concept class of either real-valued concepts or p-concepts (i.e. their corresponding conditional mean functions)

on a domain X, and let D be a distribution on X. The (un-normalized) *correlation* of two concepts $c, c' \in C$ under D is $\langle c, c' \rangle_D$.[3] The *average correlation* of C is defined to be

$$\rho_D(C) = \frac{1}{|C|^2} \sum_{c,c' \in C} \langle c, c' \rangle_D.$$

The *statistical dimension on average* at threshold γ, $\mathrm{SDA}_D(C, \gamma)$, is the largest d such that for all $C' \subseteq C$ with $|C'| \geq |C|/d$, $\rho_D(C') \geq \gamma$.

Remark 2.2.5. For any general and large concept class C (such as all one-layer neural nets), we may consider a specific subclass $C' \subseteq C$ and prove lower bounds on learning C' in terms of the SDA of C'. These lower bounds extend to C because if it is hard to learn a subset, then it is hard to learn the whole class.

We will mainly be interested in the statistical dimension in a setting where bounds on pairwise correlations are known. In that case the following lemma holds.

Lemma 2.2.6 (adapted from [FGR+17], Lemma 3.10)**.** *Suppose a concept class C has pairwise correlation γ, i.e. $\langle c, c' \rangle_D \leq \gamma$ for $c \neq c' \in C$, and squared norm at most β, i.e. $\|c\|_D^2 \leq \beta$ for all $c \in C$. Then for any $\gamma' > 0$, $\mathrm{SDA}_D(C, \gamma + \gamma') \geq |C| \frac{\gamma'}{\beta - \gamma}$. In particular, if C is a class of orthogonal concepts (i.e. $\gamma = 0$) with squared norm bounded by β, then $\mathrm{SDA}(C, \gamma') \geq |C| \frac{\gamma'}{\beta}$.*

Proof. Let $d = |C| \frac{\gamma'}{\beta - \gamma}$, and observe that for any subset $C' \subseteq C$ satisfying $|C'| \geq$

[3] In the p-concept setting, it is instructive to note that in the notation of [FGR+17], this correlation is precisely the distributional correlation $\chi_{D_0}(D_c, D_{c'})$ of the induced labeled distributions D_c and $D_{c'}$ under the reference distribution $D_0 = D \times \mathrm{Unif}\{-1, 1\}$.

$jCj/d = \frac{\beta \quad \gamma}{\gamma^0}$,

$$\begin{aligned}
\rho_D(C^0) &= \frac{1}{jC^0j^2} \sum_{c,c^0 2 C^0} jhc, c^0 i_D j \\
&\quad \frac{1}{jC^0j^2}(jC^0j\beta + (jC^0j^2 \quad jC^0j)\gamma) \\
&= \gamma + \frac{\beta \quad \gamma}{jC^0j} \\
&= \gamma + \gamma^0.
\end{aligned}$$

□

2.3 Orthogonal Family of Neural Networks

We consider neural networks with one hidden layer with activation function $\phi : \mathsf{R} \; ! \; \mathsf{R}$, and with one output node that has some activation function $\psi : \mathsf{R} \; ! \; \mathsf{R}$. If we take the input dimension to be n and the number of hidden nodes to be m, then such a neural network is a function $f : \mathsf{R}^n \; ! \; \mathsf{R}$ given by

$$f(x) = \psi\left(\sum_{i=1}^{m} a_i \phi(w_i \quad x)\right),$$

where $w_i \; 2 \; \mathsf{R}^n$ are the weights feeding into the i^{th} hidden node, and $a_i \; 2 \; \mathsf{R}$ are the weights feeding into the output node. If ψ takes values in $[\quad 1, 1]$, we may also view f as defining a p-concept in terms of its conditional mean function.

For our construction, we need our functions to be orthogonal, and we need a lower bound on their norms. For the first property we only need the distribution on the domain to satisfy a relaxed kind of spherical symmetry that we term sign-symmetry, which says that the distribution must look identical on all orthants. To lower bound the norms, we need to assume that the distribution is Gaussian $\mathsf{N}(0, I)$.

Assumption 2.3.1 (Sign-symmetry)**.** *For any $z \; 2 \; f \quad 1g^n$ and $x \; 2 \; \mathsf{R}^n$, let $x \quad z$ denote $(x_1 z_1, \ldots, x_n z_n)$. A distribution D on R^n is* sign-symmetric *if for any $z \; 2 \; f \quad 1g^n$ and x drawn from D, x and $x \quad z$ have the same distribution D.*

Assumption 2.3.2 (Odd outer activation)**.** *The outer activation ψ is an odd, increasing function, i.e. $\psi(\ \ x) = \ \ \psi(x)$.*

Note that ψ could be the identity function.

Assumption 2.3.3 (Inner activation)**.** *The inner activation ϕ 2 $L_2(\mathcal{N}(0, I))$.*

The construction of our orthogonal family of neural networks is simple and exploits sign-symmetry.

Definition 2.3.4 (Family of Orthogonal Neural Networks)**.** Let the domain be $\mathbb{R}^n$, let $\phi : \mathbb{R} \ ! \ \mathbb{R}$ be any well-behaved activation function, and let $\psi : \mathbb{R} \ ! \ \mathbb{R}$ be any odd function. For an index set $S \quad [n]$, let x_S 2 $\mathbb{R}^{jSj}$ denote the vector of x_i for i 2 S. Fix any $k > 0$. For any sign-pattern z 2 $f \ \ 1g^k$, let $\chi(z)$ denote the parity $\prod_i z_i$. For any index set $S \quad _k [n]$, define a one-layer neural network with $m = 2^k$ hidden nodes,

$$g_S(x) = \sum_{w2f \ \ 1,1g^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right)$$
$$f_S(x) = \psi\left(g_S(x)\right).$$

Our orthogonal family is

$$\mathcal{C}_{\text{orth}}(n,k) = \{f_S \ j \ S \quad _k [n]\}.$$

Notice that the size of this family is $\binom{n}{k} = n^{\ (k)}$ (for appropriate k), which is $n^{\ (\log m)}$ in terms of m. We will take $k = \Theta(\log n)$, so that $m = \text{poly}(n)$ and thus the neural networks are poly(n)-sized, and the size of the family is $n^{\ (\log n)}$, i.e. quasipolynomial in n.

We now prove that our functions are orthogonal under any sign-symmetric distribution.

Theorem 2.3.5. *Let the domain be $\mathbb{R}^n$, and let D be a sign-symmetric distribution on $\mathbb{R}^n$. Fix any $k > 0$. Then $\langle f_S, f_T \rangle_D = 0$ for any two distinct f_S, f_T 2 $\mathcal{C}_{orth}(n,k)$.*

Proof. For the proof, the key property of our construction that we will use is the following: for any sign-pattern $z \in \{\pm 1\}^n$ and any $x \in \mathbb{R}^n$,

$$f_S(x \oplus z) = \chi_S(z) f_S(x), \tag{2.3.1}$$

where $\chi_S(z) = \prod_{i \in S} z_i = \chi(z_S)$ is the parity on S of z. Indeed, observe first that

$$\begin{aligned}
g_S(x \oplus z) &= \sum_{w \in \{\pm 1\}^k} \chi(w)\phi\left(\frac{w \cdot (x \oplus z)_S}{\sqrt{k}}\right) \\
&= \sum_{w \in \{\pm 1\}^k} \chi(w)\phi\left(\frac{(w \oplus z_S) \cdot x_S}{\sqrt{k}}\right) \\
&= \sum_{w \in \{\pm 1\}^k} \chi(w \oplus z_S)\chi(z_S)\phi\left(\frac{(w \oplus z_S) \cdot x_S}{\sqrt{k}}\right) \\
&= \chi(z_S) \sum_{w \in \{\pm 1\}^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right) \qquad \text{(replacing } w \oplus z_S \text{ with } w\text{)} \\
&= \chi(z_S) g_S(x).
\end{aligned}$$

The property then follows since ψ is odd and $\psi(av) = a\psi(v)$ for any $a \in \{\pm 1\}$ and $v \in \mathbb{R}$.

Consider f_S and f_T for any two distinct $S, T \subseteq_k [n]$. Recall that by the definition of sign-symmetry, for any $z \in \{\pm 1\}^n$ and x drawn from D, x and $x \oplus z$ has the same distribution. Using this and Eq. (2.3.1), we have

$$\begin{aligned}
\langle f_S, f_T \rangle_D &= \mathop{\mathbb{E}}_{x \sim D}[f_S(x) f_T(x)] \\
&= \mathop{\mathbb{E}}_{z \in \{\pm 1\}^n} \mathop{\mathbb{E}}_{x \sim D}[f_S(x \oplus z) f_T(x \oplus z)] && \text{(sign-symmetry)} \\
&= \mathop{\mathbb{E}}_{z \in \{\pm 1\}^n} \mathop{\mathbb{E}}_{x \sim D}[\chi_S(z) f_S(x) \chi_T(z) f_T(x)] && \text{(Eq. (2.3.1))} \\
&= \mathop{\mathbb{E}}_{x \sim D}\left[f_S(x) f_T(x) \mathop{\mathbb{E}}_{z \in \{\pm 1\}^n} \chi_S(z)\chi_T(z)\right] \\
&= 0,
\end{aligned}$$

since $\mathbb{E}_{z \in \{\pm 1\}^n} \chi_S(z)\chi_T(z) = 0$ for any two distinct parities χ_S, χ_T. □

Remark 2.3.6. Our proof actually shows that any family of functions satisfying Eq. (2.3.1) is an orthogonal family under any sign-symmetric distribution.

We still need to establish that our functions are nonzero. For this we need to specialize to the Gaussian distribution, as well as consider specific activation functions (a similar analysis can in principle be carried out for other sign-symmetric distributions). For any n and k, it follows from Lemma A.1.1 that if the inner activation ϕ has a nonzero Hermite coefficient of degree k or higher, then the functions in $C_{\text{orth}}(n,k)$ are nonzero. The sigmoid, ReLU and sign functions all satisfy this property.

Corollary 2.3.7. *Let the domain be $\mathbb{R}^n$, and let D be any sign-symmetric distribution on $\mathbb{R}^n$. For any $\gamma > 0$,*

$$\mathrm{SDA}_D(C_{orth}(n,k),\gamma) \quad |C_{orth}(n,k)|\gamma = \binom{n}{k}\gamma.$$

Here we also assume that all $c \in C_{orth}(n,k)$ are nonzero for our distribution D.

Proof. Follows from Theorem 2.3.5 and Lemma 2.2.6, using a loose upper bound of 1 on the squared norm. □

We also need to prove norm lower bounds on our functions for our notions of learning to be meaningful. In Section A.1, we prove the following.

Theorem 2.3.8. *Let the inner activation function ϕ be* ReLU *or sigmoid, and let the outer activation function ψ be any odd, increasing, continuous function. Let the underlying distribution D be $N(0,I_n)$. Then $\|f_S\| = \Omega(e^{(k)})$, where the hidden constants depend on ψ and ϕ, for any $f_S \in C_{orth}(n,k)$.*

With this in hand, we now state our main SQ lower bounds.

Theorem 2.3.9. *Let the input dimension be n, and let the underlying distribution be $N(0,I_n)$. Consider $C_{orth}(n,k)$ instantiated with $\phi =$* ReLU *or sigmoid and ψ any odd, increasing function (including the identity function), and let $m = 2^k$ be the hidden*

layer size of each neural net. Let A be an SQ learner using only inner product queries of tolerance τ. For any k 2 N, there exists $\tau = 1/n^{(k)}$ such that A requires at least $n^{(k)}$ queries of tolerance τ to learn $C_{orth}(n,k)$ with advantage $1/\exp(k)$.

In particular, there exist $k = \Theta(\log n)$ and $\tau = 1/n^{(\log n)}$ such that A requires at least $n^{(\log n)}$ queries of tolerance τ to learn $C_{orth}(n,k)$ with advantage $1/\operatorname{poly}(n)$. In this case $m = \operatorname{poly}(n)$, so that each function in the family has polynomial size. This is our main superpolynomial lower bound.

Proof. The proof amounts to careful choices of the parameters ϵ, γ and τ in Corollary 2.3.7 and Corollary 2.4.6. Recall that $\mathrm{SDA}(C_{\mathrm{orth}}(n,k),\gamma) \quad n^{(k)}\gamma$. We pick $\gamma = n^{(k)}$ appropriately such that $d = \mathrm{SDA}(C_{\mathrm{orth}}(n,k),\gamma)$ is still $n^{(k)}$. Theorem 2.3.8 gives us a norm lower bound of $\exp(\Theta(k))$, allowing us to take $\epsilon = \exp(\Theta(k))$ and $\tau = \sqrt{\gamma} = n^{(k)}$ in Corollary 2.4.6. □

2.4 SQ Lower Bounds

2.4.1 SQ Lower Bounds for Real-valued Functions

Prior work [Szö09, Fel12] has already established the following fundamental result, which we phrase in terms of our definition of statistical dimension. For the reader's convenience, we include a proof in Section A.2.

Theorem 2.4.1. *Let D be a distribution on X, and let C be a real-valued concept class over a domain X such that $\|c\|_D > \epsilon$ for all c 2 C. Consider any SQ learner that is allowed to make only inner product queries to an SQ oracle for the labeled distribution D_c for some unknown c 2 C. Let $d = \mathrm{SDA}_D(C,\gamma)$. Then any such SQ learner needs at least $\Omega(d)$ queries of tolerance $\sqrt{\gamma}$ to learn C up to L_2 error ϵ.*

2.4.2 SQ Lower Bounds for *p*-concepts

It turns out to be fruitful to view our learning problem in terms of a decision problem over distributions. We define the problem of distinguishing a valid labeled

distribution from a randomly labeled one, and show a lower bound for this problem. We then show that learning is at least as hard as distinguishing, thereby extending the lower bound to learning as well. Our analysis closely follows that of [FGR+17].

Definition 2.4.2 (Distinguishing between labeled and uniformly random distributions)**.** Let C be a class of p-concepts over a domain X, and let D be a distribution on X. Let $D_0 = D_{c_0}$ be the randomly labeled distribution D Unif$f$ 1g. Suppose we are given SQ access either to a labeled distribution D_c for some c 2 C such that c 6= c_0 or to D_0. The problem of distinguishing between labeled and uniformly random distributions is to decide which.

Remark 2.4.3. Given access to D_c for some truly boolean concept $c : X$! f 1g, it is easy to distinguish any other boolean function c' from c since kc $c'k_D^2$ = 2 2$hc, c'i_D$ (which is information-theoretically optimal as a distinguishing criterion) can be computed using a single inner product query. However, if c and c' are p-concepts, kck_D and $kc'k_D$ are not 1 in general and may be difficult to estimate. It is not obvious how best to distinguish the two, short of directly learning the target.

Considering the distinguishing problem is useful because if we can show that distinguishing itself is hard, then any reasonable notion of learning will be hard as well, including weak learning. We give simple reductions for both our notions of learning.

Lemma 2.4.4 (Learning is as hard as distinguishing)**.** *Let D be a distribution over the domain X, and let C be a p-concept class over X. Suppose there exists either*

(a) a weak SQ learner capable of learning C up to advantage ϵ using q queries of tolerance τ, where τ $\epsilon/2$; or,

(b) an SQ learner capable of learning C (assume kck_D 3ϵ for all c 2 C) up to L_2 error ϵ using q queries of tolerance τ, where τ ϵ^2. Then there exists a distinguisher that is able to distinguish between an unknown D_c and D_0 using at most $q + 1$ queries of tolerance τ.

Proof. (a) Run the weak learner to obtain $\tilde{c}$. If $c \neq c_0$, we know that $\langle \tilde{c}, c \rangle_D \leq \epsilon$, whereas if $c = c_0$, then $\langle \tilde{c}, c \rangle_D = 0$ no matter what $\tilde{c}$ is. A single additional query $(h(x,y) = \tilde{c}(x)y)$ of tolerance $\epsilon/2$ distinguishes between the two cases.

(b) Run the learner to obtain $\tilde{c}$. If $c \neq c_0$, i.e. $\|c - c_0\|_D \geq 3\epsilon$, we know that $\|\tilde{c} - c\|_D \geq \epsilon$, so that by the triangle inequality, $\|\tilde{c} - c_0\|_D \geq \|c - c_0\|_D - \|\tilde{c} - c\|_D \geq 2\epsilon$. But if $c = c_0$, then $\|\tilde{c} - c_0\|_D \leq \epsilon$. An additional query $(h(x,y) = \tilde{c}(x)^2)$ of tolerance ϵ^2 suffices to distinguish the two cases. □

We now prove the main lower bound on distinguishing.

Theorem 2.4.5. *Let D be a distribution over the domain X, and let C be a p-concept class over X. Then any SQ algorithm needs at least $d = \mathrm{SDA}(C, \gamma)$ queries of tolerance $p\overline{\gamma}$ to distinguish between D_c and D_0 for an unknown $c \in C$. (We will consider deterministic SQ algorithms that always succeed, for simplicity.)*

Proof. Consider any successful SQ algorithm A. Consider the adversarial strategy where to every query $h : X \to \{-1, 1\} \to [-1, 1]$ of A (with tolerance $\tau = p\overline{\gamma}$), we respond with $\mathsf{E}_{D_0}[h]$. We can pretend that this is a valid answer with respect to any $c \in C$ such that $|\mathsf{E}_{D_c}[h] - \mathsf{E}_{D_0}[h]| \leq \tau$. Our argument will be based on showing that each such query rules out fairly few distributions, so that the number of queries required in total is large.

Since we assumed that A is a deterministic algorithm that always succeeds, it eventually correctly guesses that it is D_0 that it is getting answers from. Say it takes q queries to do so. For the k^{th} query h_k, let S_k be the set of concepts in C that are ruled out by our response $\mathsf{E}_{D_0}[h_k]$:

$$S_k = \{c \in C \mid |\mathop{\mathsf{E}}_{D_c}[h] - \mathop{\mathsf{E}}_{D_0}[h]| > \tau\}.$$

We'll show that

(a) on the one hand, $\bigcup_{k=1}^{q} S_k = C$, so that $\sum_{k=1}^{q} |S_k| \geq |C|$,

(b) while on the other, $|S_k| \quad |C|/d$ for every k. Together, this will mean that $q \quad d$.

For the first claim, suppose $\bigcup_{k=1}^{q} S_k$ were not all of C, and indeed say $c \in C \setminus (\bigcup_{k=1}^{q} S_k)$. This is a distribution that our answers were consistent with throughout, yet one that A's solution (D_0) is incorrect for. But A always succeeds, so for it not to have ruled out this D_c is impossible.

For the second claim, suppose for the sake of contradiction that for some k, $|S_k| > |C|/d$. By Definition 2.2.4, this means we know that $\rho_D(S_k) \quad \gamma$. One of the key insights in the proof of [Szö09] is that by expressing query expectations entirely in terms of inner products, we gain the ability to apply simple algebraic techniques. To this end, for any query function h, let $\widehat{h}(x) = (h(x,1) \quad h(x, \quad 1))/2$. Observe that for any p-concept c,

$$\begin{aligned}
\langle \widehat{h}, c \rangle_D &= \mathop{\mathbb{E}}_{x \ D}\left[h(x,1)\frac{c(x)}{2}\right] \quad \mathop{\mathbb{E}}_{x \ D}\left[h(x, \quad 1)\frac{c(x)}{2}\right] \\
&= \mathop{\mathbb{E}}_{x \ D}\left[h(x,1)\frac{1+c(x)}{2}\right] \\
&\quad + \mathop{\mathbb{E}}_{x \ D}\left[h(x, \quad 1)\frac{1 \quad c(x)}{2}\right] \\
&\quad \mathop{\mathbb{E}}_{x \ D}\left[h(x,1)\frac{1}{2}\right] \quad \mathop{\mathbb{E}}_{x \ D}\left[h(x, \quad 1)\frac{1}{2}\right] \\
&= \mathop{\mathbb{E}}_{D_c}[h] \quad \mathop{\mathbb{E}}_{D_0}[h],
\end{aligned}$$

the difference between the query expectations wrt D_c and D_0. Here we have expanded each $\mathbb{E}_{D_c}[h]$ using the fact that the label for x is 1 with probability $(1 + c(x))/2$ and 1 otherwise. Thus $|\langle h, \widehat{h_k}, c \rangle_D|$, where h_k is the k^{th} query, is greater than τ for any $c \in S_k$, since S_k are precisely those concepts ruled out by our response. We will show contradictory upper and lower bounds on the following quantity:

$$\Phi = \left\langle \widehat{h_k}, \sum_{c \in S_k} c \ \ \text{sign}(\langle h, \widehat{h_k}, c \rangle_D) \right\rangle_D .$$

Note that since every query h satisfies $\|h(\,,y)\|_D \quad 1$ for all y, it follows by the triangle inequality that $\|\widehat{h}\|_D \quad 1$. So by Cauchy-Schwarz and our observation that $\rho_D(S_k) \quad \gamma$,

$$\Phi^2 \quad \|\widehat{h_k}\|_D^2 \quad \left\|\sum_{c2S_k} c \quad \text{sign}(h\widehat{h_k}, c)\right\|_D^2$$
$$\sum_{c,c'2S_k} |hc, c'i_D| = |S_k|^2 \rho_D(S_k) \quad |S_k|^2\gamma.$$

However since $|h\widehat{h_k}, ci_D| \; > \tau$, we also have that $\Phi = \sum_{c2S_k} |h\widehat{h_k}, ci_D| \; > |S_k|\tau$. Since $\tau = p_{\overline{\gamma}}$, this contradicts our upper bound and in turn completes the proof of our second claim. And as noted earlier, the two claims together imply that $q \quad d$. □

The final lower bounds on learning thus obtained are stated as a corollary for convenience. The proof follows directly from Lemma 2.4.4 and Theorem 2.4.5.

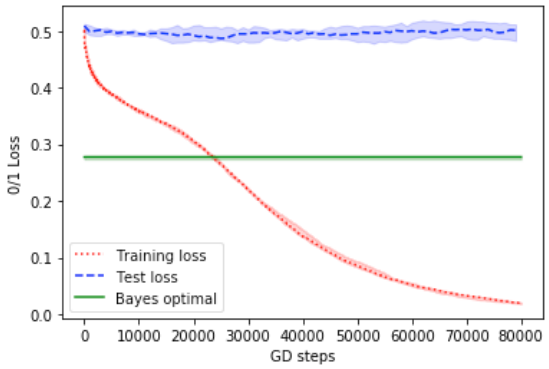
Corollary 2.4.6. *Let D be a distribution over the domain X, and let C be a p-concept class over X. Let γ, τ be such that $p_{\overline{\gamma}} \quad \tau$. Let $d = \text{SDA}(C, \gamma)$.*

(a) Let ϵ be such that $\tau \quad \epsilon^2$, and assume $\|c\|_D \quad 3\epsilon$ for all c 2 C. Then any SQ learner learning C up to L_2 error ϵ requires at least $d \quad 1$ queries of tolerance τ.

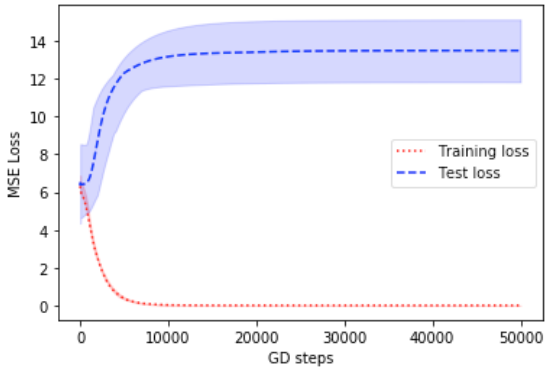
(b) Let ϵ be such that $\tau \quad \epsilon/2$. Then any weak SQ learner learning C up to advantage ϵ requires at least $d \quad 1$ queries of tolerance τ.

2.5 Experiments

We include experiments for both regression and classification. We train an overparameterized neural network on data from our function class, using gradient descent. We find that we are able to achieve close to zero training error, while test error remains high. This is consistent with our lower bound for these classes of functions.



(a) Learning a softmax of a one-layer tanh network



(b) Learning a linear combination of tanhs

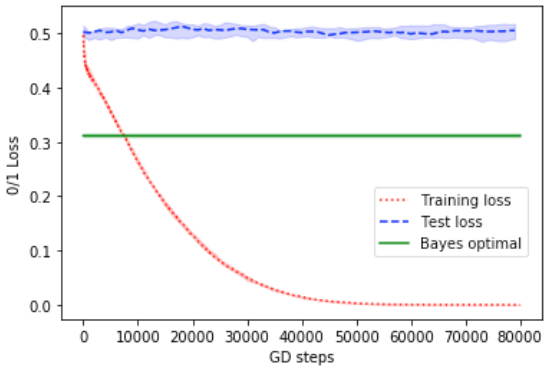
Figure 2.1: In (a) the target function is a softmax (1 labels) of a sum of 2^7 tanh activations with $n = 14$; in (b) the labels are obtained similarly but without the softmax. In both cases, we train a 1-layer neural network with 5 $2^7 = 640$ tanh units (hence 10241 parameters) using a training set of size 6000 and a test set of size 1000, with the learning rate set to 0.01. For (a) we take the sign of this trained network and measure its training and testing 0/1 loss; for (b) we measure the train and test square-loss of the learned network directly. In (a) we also plot the test error of the bayes optimal network (sign of the target function).

For classification, we use a training set of size T of data corresponding to $f \in C_{\text{orth}}(n,k)$ instantiated with $\phi = \tanh$ and $\psi = \tanh$. We draw $x \sim N(0, I_n)$. For each x, y is picked randomly from $\{\pm 1\}$ in such a way that $\mathbb{E}[y|x] = f(x)$. Since the outer activation ψ is tanh, this can be thought of as applying a softmax to the network's output, or as the Boolean label corresponding to a logit output. We train a sum of tanh network (i.e. a network in which the inner activation is tanh and no outer activation is applied) on this data using gradient descent on squared loss, threshold the output, and plot the resulting 0/1 loss. See Fig. 2.1(a). This setup models a common way in which neural networks are trained for classification problems in practice.

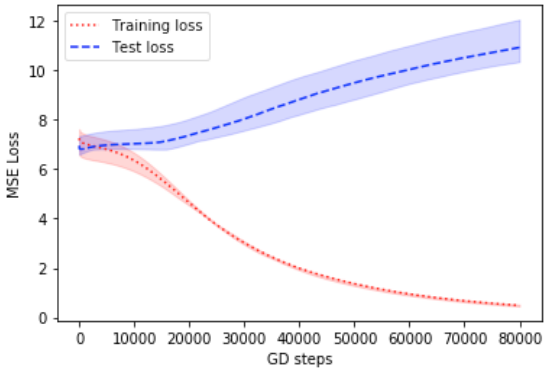
For regression, we use a training set of size T of data corresponding to $f \in C_{\text{orth}}(n,k)$ instantiated with $\phi = \tanh$ and ψ being the identity. We draw $x \sim N(0, I_n)$, and $y = f(x)$. We train a sum of tanh network on this data using gradient descent on squared loss, which we plot in Fig. 2.1(b). This setup models the natural way of using neural networks for regression problems.

In both cases, we train neural networks whose number of parameters considerably exceeds the amount of training data. In all our experiments, we plot the median over 10 trials and shade the inter-quartile range of the data.

Similar results hold with the inner activation ϕ being ReLU instead of tanh, and are shown in Fig. 2.2.



(a) Learning a softmax of a one-layer ReLU network



(b) Learning a linear combination of ReLUs

Figure 2.2: In (a) the target function is a softmax (1 labels) of a sum of 2^8 ReLU activations with $n = 14$; in (b) the labels are obtained similarly but without the softmax. In both cases, we train a 1-layer neural network with 5 $2^8 = 1280$ ReLU units (hence 20481 parameters) using a training set of size 6000 and a test set of size 1000, with the learning rate set to 0.005 for classification and 0.002 for regression. For (a) we take the sign of this trained network and measure its training and testing 0/1 loss; for (b) we measure the train and test square loss of the learned network directly. In (a) we also plot the test error of the bayes optimal network (sign of the target function).

Chapter 3: Hardness of Noise-Free Learning for Two-Hidden-Layer Neural Networks

3.1 Introduction

In this chapter we extend a central line of research proving representation-independent hardness results for learning classes of neural networks. We will consider arguably the simplest possible setting: given samples $(x_1, y_1), \ldots, (x_n, y_n)$ where for every i 2 $[n]$, x_i is sampled independently from some distribution D over R^d and $y_i = f(x_i)$ for an unknown neural network $f : \mathsf{R}^d$! R, the goal is to output any function $\widehat{f}$ for which $\mathsf{E}_{x \ D}[(f(x) \quad \widehat{f}(x))^2]$ is small. This model is often referred to as the *realizable* or *noise-free* setting.

This problem has long been known to be computationally hard for discrete input distributions. For example, if D is supported over a discrete domain like the Boolean hypercube, then we have a variety of hardness results based on cryptographic/average-case assumptions [KS09, DLSS14, Dan16a, DV20b, DV21].

Over the last few years there has been a very active line of research on the complexity of learning with respect to continuous distributions, the most widely studied case being the assumption that D is a standard Gaussian in d dimensions. A rich algorithmic toolbox has been developed for the Gaussian setting [JSA15, ZSJ+17b, BG17b, LY17, Tia17, GKM18, GLM18b, BJW19, ZYWG19b, DGK+20, LMZ20, DK20, ATV21, CKM20, SZB21, VSS+22], but all known efficient algorithms can only handle networks with a single hidden layer, that is, functions of the form $f(x) = \sum_{i=1}^{k} \lambda_i \sigma(\hbar w_i, x i)$. This motivates the following well-studied question:

Are there fundamental barriers to learning neural networks with two *hidden layers?* (3.1.1)

Two distinct lines of research, one using cryptography and one using the statistical query (SQ) model, have made progress towards solving this question.

In the cryptographic setting, [DV21] showed that the existence of a certain class of pseudorandom generators, specifically local pseudorandom generators with polynomial stretch, implies superpolynomial lower bounds for learning ReLU networks with *three* hidden layers.

For SQ learning, work of [GGJ+20] and [DKKZ20] gave the first superpolynomial *correlational* SQ (CSQ) lower bounds for learning even one-hidden-layer neural networks. Notably, however, there are strong separations between SQ and CSQ [APVZ14, ADHV19, CKM20], and the question of whether a general SQ algorithm exists remained an interesting open problem. In fact, Vempala and Wilmes [VW19a] showed that general SQ lower bounds might be impossible to achieve for learning real-valued neural networks. For any family of networks satisfying a simple non-degeneracy condition (see Section 3.1.1), they gave an algorithm that succeeded using only polynomially many statistical queries. As such, the prevailing conventional wisdom was that noise was required in the model to obtain full SQ lower bounds.

The main contribution of this chapter is to answer Question 3.1.1 by giving both general SQ lower bounds and cryptographic hardness results (based on the Learning with Rounding or LWR assumption) for learning ReLU networks with two hidden layers and polynomially bounded weights.[1] We note that our SQ lower bound is the first of its kind for learning ReLU networks of *any* depth. We also show how to extend our results to the setting where the learner has label query access to the unknown network.

SQ Lower Bound We state an informal version of our main SQ lower bound:

Theorem 3.1.1 (Full SQ lower bound for two hidden layers (informal), see Theorem 3.4.1)**.** *Any SQ algorithm for learning* $\mathrm{poly}(d)$*-sized two-hidden-layer ReLU net-*

[1] Note that if the weights were allowed to be arbitrarily large, it is well-known to be trivial to obtain hardness over Gaussian inputs from hardness over Boolean inputs: simply approximate the sign function arbitrarily well and convert all but an arbitrarily small fraction of Gaussian inputs to bitstrings.

Reference	Num. hidden layers	Model of hardness
[DKKZ20, GGJ+20]	1	Correlational SQ
[DV21]	3	Cryptographic (assuming existence of local PRGs)
This work	2	Full SQ
This work	2	Cryptographic (assuming hardness of LWR)

Table 3.1: Summary of known and new superpolynomial lower bounds for learning noise-free shallow ReLU networks over Gaussian inputs up to sufficiently small (but non-negligible) error. (Definitions and terminology may be found in Section 3.2.)

works over $N(0, \mathrm{Id}_d)$ *up to squared loss* $1/\operatorname{poly}(d)$ *must use at least* $d^{\omega(1)}$ *queries, or have query tolerance that is negligible in* d*.*

We stress that this bound holds unconditionally, independent of any cryptographic assumptions. This simultaneously closes the gap between the hardness result of [DV21] and the positive results on one-hidden-layer networks [JSA15, ZSJ+17b, GLM18b, ATV21, DK20] and goes against the conventional wisdom that one cannot hope to prove full SQ lower bounds for learning real-valued functions in the realizable setting.

We also note that unlike previous CSQ lower bounds which are based on orthogonal function families and crucially exploit cancellations specific to the Gaussian distribution, our Theorem 3.1.1 and other hardness results in this chapter easily extend to any reasonably anticoncentrated and symmetric product distribution over R^d; see Remark 3.3.11.

Cryptographic Lower Bound While Theorem 3.1.1 rules out almost all known approaches for provably learning neural networks (e.g. method of moments/tensor decomposition [JSA15, ZSJ+17b, GLM18b, BJW19, DGK+20, DK20, ATV21], noisy gradient descent [BG17b, LY17, Tia17, GKM18, ZYWG19b, LMZ20], and filtered

PCA [CKM20]), it does not preclude the existence of a non-SQ algorithm for doing so. Indeed, a number of recent works [BRST21, SZB21, ZSWB22, DK21] have ported algorithmic techniques like lattice basis reduction [LLL82], traditionally studied in the context discrete settings like cryptanalysis, to learning problems over continuous domains for which there is no corresponding SQ algorithm.

Our next result shows however that under a certain cryptographic assumption, namely hardness of *Learning with Rounding (LWR) with polynomial modulus* [BPR12, AKPW13, BGM+16] (see Section 3.2), no polynomial-time algorithm can learn two-hidden-layer neural networks from Gaussian examples.

Theorem 3.1.2 (Cryptographic hardness result (informal), see Theorem 3.5.1)**.** *Suppose there exists a* $\mathrm{poly}(d)$*-time algorithm for learning* $\mathrm{poly}(d)$*-sized two-hidden-layer ReLU networks over* $N(0, \mathrm{Id}_d)$ *up to squared loss* $1/\mathrm{poly}(d)$*. Then there exists a quasipolynomial-time algorithm for LWR with polynomial modulus.*

Note that here we may actually improve the LWR hardness assumption required from quasipolynomial to any mildly superpolynomial function of the security parameter (see Remark 3.5.2).

Under LWR with polynomial modulus, we also show the first hardness result for learning *one hidden layer* ReLU networks over the uniform distribution on $\{0,1\}^d$ (see Theorem 3.5.3).

In Section 3.2, we discuss existing hardness evidence for LWR as well as its relation to more standard assumptions like Learning with Errors. From a negative perspective, Theorem 3.1.2 suggests that the aforementioned lattice-based algorithms for continuous domains are unlikely to yield new learning algorithms for two-hidden-layer networks, because even their more widely studied *discrete* counterparts have yet to break LWR. From a positive perspective, in light of the prominent role LWR and its variants have played in a number of practical proposals for post-quantum cryptography [CKLS18, BGML+18, JZ16, DKRV18], Theorem 3.1.2 offers a new avenue for stress-testing these schemes.

Query Learning Lower Bound One additional benefit of our techniques is that they are flexible enough to accommodate other learning models beyond traditional PAC learning. To illustrate this, for our final result we show hardness of learning neural networks from *label queries*. In this setting, the learner is much more powerful: rather than sample or SQ access, they are given the ability to query the value $f(x)$ of the unknown function f at any desired point x in R^d, and the goal is still to output a function $\widehat{f}$ for which $\mathsf{E}[(f(x) \quad \widehat{f}(x))^2]$ is small. The expectation here is with respect to some specified distribution, which we will take to be $N(0, \mathrm{Id}_d)$, though as before, our techniques will apply to any reasonably anticoncentrated, symmetric product distribution over R^d.

In recent years, this question has received renewed interest from the security and privacy communities in light of *model extraction attacks*, which attempt to reverse-engineer neural networks found in publicly deployed systems [TJ+16, MSDH19, PMG+17, JCB+20, RK20, JWZ20, DG21]. Recent work [CKM21] has shown that in this model, there is an efficient algorithm for learning arbitrary one-hidden-layer ReLU networks that is truly polynomial in all relevant parameters. We show that under plausible cryptographic assumptions about the existence of simple pseudorandom function (PRF) families (see Section 3.6) which may themselves be based on standard number theoretic or lattice-based cryptographic assumptions, such a guarantee is impossible for general *constant-depth* ReLU networks.

Theorem 3.1.3 (Label query hardness (informal), see Theorem 3.6.1)**.** *If either the decisional Di e{Hellman or the Learning with Errors assumption holds, then the class of* $\mathrm{poly}(d)$*-sized constant-depth ReLU networks from* R^d *to* R *is not learnable up to small constant squared loss* ϵ *over* $N(0, \mathrm{Id}_d)$ *even using label queries over all of* R^d*.*

Note that the connection between PRFs and hardness of learning from label queries over *discrete domains* is a well-known connection dating back to Valiant [Val84].

To our knowledge, however, Theorem 3.1.3 is the first hardness result for query learning over continuous domains.

3.1.1 Discussion and Related Work

Hardness for learning neural networks. There are a number of works [BR89, Vu06, KS09, LSSS14, GKKT17a, DV20b] showing hardness for *distribution-free* learning of various classes of neural networks.

As for hardness of distribution-specific learning, several works have established lower bounds with respect to the Gaussian distribution. Apart from the works [GGJ+20, DKKZ20, DV21] from the introduction which are most closely related to the present work, we also mention the works of [KK14a, GKK19, GGK20, DKZ20a] which showed hardness for *agnostically* learning halfspaces and ReLUs, [Sha18b] which showed hardness for learning periodic activations with gradient-based methods, [SVWX17] which showed lower bounds against SQ algorithms for learning one-hidden-layer networks using *Lipschitz* statistical queries and large tolerance, and [SZB21] which showed lattice-based hardness of learning one-hidden-layer networks when the labels y_i have been perturbed by bounded *adversarially chosen* noise. Our approach has similarities to the "Gaussian lift" as studied by Klivans and Kothari [KK14a]. Their approach, however, required noise in the labels, whereas we are interested in hardness in the strictly *realizable setting*. We also remark that [DGKP20, AAK21] showed *correlational* SQ lower bounds for learning random depth-$\omega(\log n)$ neural networks over *Boolean inputs* which are uniform over a halfspace.

There have also been works on hardness of learning from label queries over *discrete domains* and for more "classical" concept classes like Boolean circuits [Fel09, CGV15, Val84, Kha95, AK95].

Lastly, we remark on how our results relate to [CKM20], which gives the only known upper bound for learning neural networks over Gaussian inputs beyond one hidden layer. They showed that learning ReLU networks of arbitrary depth is "fixed-

parameter tractable" in the sense that there is a fixed function $g(k, \epsilon)$ in the size k of the network and target error ϵ for which the time complexity is at most $g(k, \epsilon)$ poly(d), and their algorithm can be implemented in SQ. That said, this does not contradict our lower bounds for two reasons: 1) their algorithm only applies to networks without biases, 2) in our lower bound constructions, k scales polynomially in d.

SQ lower bounds for real-valued functions. A recurring conundrum in the literature on SQ lower bounds for supervised learning has been whether one can show SQ hardness for learning *real-valued* functions. SQ lower bounds for Boolean functions are typically shown by lower bounding the *statistical dimension* of the function class, which essentially corresponds to the largest possible set of functions in the class which are all approximately pairwise orthogonal. Indeed, the content of the hardness results of [GGJ+20, DKKZ20] was to prove lower bounds on the statistical dimension of one-hidden-layer networks. Unfortunately, for real-valued functions, statistical dimension lower bounds only imply CSQ lower bounds. As discussed in [GGJ+20], the class of d-variate Hermite polynomials of degree-ℓ is pairwise orthogonal and of size $d^{O(\ell)}$, which translates to a CSQ lower bound of $d^{\Omega(\ell)}$. Yet there exist SQ algorithms for learning Hermite polynomials in far fewer queries [APVZ14, ADHV19].

Further justification for the difficulty of proving SQ lower bounds for real-valued functions came from [VW19a], which observed that for any real-valued learning problem satisfying a seemingly innocuous non-degeneracy assumption—namely that for any pair of functions f, g in the class, the probability under the input distribution $\boldsymbol{D}$ that $f(x) = g(x)$ is zero—there is an efficient "cheating" SQ algorithm (see Proposition 4.1 therein). The SQ lower bound shown in the present work circumvents this proof barrier by exhibiting a family of neural networks for which any pair of networks agrees on a set of inputs with Gaussian measure *bounded away from zero*.

Open questions While our results settle Question 3.1.1, a number of intriguing gaps between our lower bounds and existing upper bounds remain open:

- **General one-hidden-layer networks.** Despite the considerable amount of work on learning one-hidden-layer networks over Gaussian inputs, all known positive results that run in polynomial time in all parameters (input dimension d, network size k, inverse error $1/\epsilon$) still need to make various assumptions on the structure of the network. Remarkably, it is even open whether one-hidden-layer ReLU networks with *positive output layer weights* (i.e. "sums of ReLUs") can be learned in polynomial time, the best known guarantee being the $(k/\epsilon)^{\log^2 k}$ $\mathrm{poly}(d/\epsilon)$-time algorithm of [DK20]. As for general one-hidden-layer ReLU networks, it is still open whether they can even be learned in time $d^{O(k)}$ $\mathrm{poly}(1/\epsilon)$, the best known guarantee being the $k^{\mathrm{poly}(k/\epsilon)}$ $\mathrm{poly}(d)$-time algorithm of [CKM20].

- **Query learning shallow networks.** While Theorem 3.1.3 establishes that above a certain constant depth, ReLU networks cannot be learned even from label queries over the Gaussian distribution. It would be interesting to close the gap between this and the positive result of [CKM21] which only applies to one-hidden-layer networks, although fully settling this seems closely related to the question of what are the shallowest possible Boolean circuits needed to implement pseudorandom functions, a longstanding open question in circuit complexity.

3.1.2 Technical Overview

Our work will build on a recent approach of Daniely and Vardi [DV21], who developed a simple and clever technique for lifting discrete functions to the Gaussian domain entirely in the realizable setting. Our main contributions are to (1) make their lifting procedure more efficient so that two hidden layers suffice and (2) show how to apply the lift in a variety of models beyond PAC. For the purposes of this overview we will take the domain of our discrete functions to be $\{0,1\}^d$, but our techniques extend to $\mathbb{Z}_q^d$ with $q = \mathrm{poly}(d)$.

Daniely–Vardi (DV) lift. At a high level, the DV lift is a transformation mapping a Boolean example (x, y) labeled by a hard-to-learn Boolean function f to a Gaussian example $(z, \widetilde{y})$ labeled by a (real-valued) ReLU network f^{DV} that behaves similarly to f in that $f^{\mathsf{DV}}(z)$ approximates $f(\mathrm{sign}(z))$, where for us $\mathrm{sign}(t)$ denotes $\mathbb{1}[t > 0]$ and is applied elementwise. The key idea is to use a continuous approximation $\widetilde{\mathrm{sign}}$ of the sign function, and to pair it with a "soft indicator" function $\mathrm{bad} : \mathsf{R}^d \to \mathsf{R}_+$ that is large whenever $\mathrm{sign}(z) \neq \widetilde{\mathrm{sign}}(z)$, and that can be implemented as a one-hidden-layer network independent of the target function. One can show that whenever f is realizable as an L-hidden-layer network over $\{0,1\}^d$, the function $f^{\mathsf{DV}}(z) = \mathrm{ReLU}(f(\widetilde{\mathrm{sign}}(z)) - \mathrm{bad}(z))$ can be implemented as an $(L+2)$-hidden-layer network satisfying

$$f^{\mathsf{DV}}(z) = \mathrm{ReLU}(f(\mathrm{sign}(z)) - \mathrm{bad}(z)). \tag{3.1.2}$$

This property allows us to generate synthetic Gaussian labeled examples $(z, f^{\mathsf{DV}}(z))$ from Boolean labeled examples $(x, f(x))$, and thereby reduce the problem of learning f to that of learning f^{DV}. For a fuller overview, see Section 3.3.1.

Improving the DV lift. Our first technical contribution is to introduce a more efficient lift which only requires one extra hidden layer. Our starting point is to observe that a variety of hard-to-learn Boolean functions f like parity and LWR take the form $f(x) = \sigma(h(x))$ for some ReLU network h whose range T over Boolean inputs is a discrete subset of $[0, \mathrm{poly}(d)]$ of *polynomially bounded* size, and for some function $\sigma : T \to [0,1]$. For such *compressible* functions (see Definition 3.3.1), one can write $f(x) = \sigma(h(x)) = \sum_{t \in T} \sigma(t)\mathbb{1}[h(x) = t]$. Again, we would like to implement lifted function $f^{\mathsf{M}} : \mathsf{R}^d \to \mathsf{R}$ using $\widetilde{\mathrm{sign}}$ and bad so that it approximates $f(\mathrm{sign}(z))$ except when bad indicates that $\widetilde{\mathrm{sign}} \neq \mathrm{sign}$. To this end, we might hope to implement, say,

$$f^{\mathsf{M}}(z) = \sum_{t \in T} \sigma(t)\mathbb{1}[h(\widetilde{\mathrm{sign}}(z)) = t]\mathbb{1}[\exists j : \mathrm{bad}(z_j) \geq 1].$$

Here we now view bad as a univariate function, and whenever it is small, we can be sure $\widetilde{\mathrm{sign}} = \mathrm{sign}$. Suppose that we could build a one-hidden-layer network $N(s_1, \ldots, s_d; t)$

that behaves like $\mathbb{1}[t = 0]\mathbb{1}[\mathit{8}j : s_j \quad 1]$. Then we could realize f^{M} as an $(L+1)$-hidden-layer network:

$$f^{\mathsf{M}}(z) = \sum_{t \ 2T} \sigma(t \ )N(\mathrm{bad}(z_1), \ldots, \mathrm{bad}(z_d);\ h(\widetilde{\mathrm{sign}}(z)) \quad t \ ). \tag{3.1.3}$$

While many natural attempts to build such an N run into difficulties, we construct a suitably relaxed version of N that turns out to suffice for the reduction. To gain some intuition for our construction, the starting observation is that the following inclusion-exclusion type formula vanishes identically whenever any of the s_j is 1:

$$\psi(s_1, s_2, s_3) \quad \psi(1, s_2, s_3) \quad \psi(s_1, 1, s_3) \quad \psi(s_1, s_2, 1) \tag{3.1.4}$$

$$+ \psi(s_1, 1, 1) + \psi(1, s_2, 1) + \psi(s_1, 1, 1) \quad \psi(1, 1, 1). \tag{3.1.5}$$

For a suitable choice of ψ, one might hope to build N out of such a formula by taking $s_j = \mathrm{bad}(z_j)$ for every j. But the natural generalization of this expression to d inputs would have size 2^d, which runs the risk of rendering the resulting SQ lower bounds vacuous. Our final construction (Lemma 3.3.10) instead resembles a truncated inclusion-exclusion type formula of only quasipolynomial size, which may be of independent interest. Since the SQ lower bounds for Boolean functions that we build on are exponential, by a simple padding argument we still obtain a superpolynomial SQ lower bound for our lifted functions.

Hard one-hidden-layer Boolean functions and LWR. To use this lift for Theorems 3.1.1 and 3.1.2, we need one-hidden-layer networks that are *compressible* and hard to learn over uniform Boolean inputs. For SQ lower bounds, we can simply start from parities, for which there are exponential SQ lower bounds, and which turn out to be easily implementable by compressible one-hidden-layer networks. For cryptographic hardness, Daniely and Vardi [DV21] used certain one-hidden-layer Boolean networks that arise from the cryptographic assumption that local PRGs exist (see Section A.4.1 therein). Unfortunately, these functions are not compressible. For

this reason, we work instead with LWR: it turns out that the LWR functions are compressible and, conveniently, the hardness assumption directly involves uniform discrete inputs.

Hardness beyond PAC. While the DV lift is *a priori* only for showing hardness of example-based PAC learning, we can extend it to the SQ and label query models by simple simulation arguments.

3.2 Preliminaries

3.2.1 Notation

We use Unif(S) to denote the uniform distribution over a set S. We use U_d as shorthand for Unif$\{0,1\}^d$. We use $\mathcal{N}(0, \mathrm{Id}_d)$ (or sometimes $\mathcal{N}_d$ for short) to denote the standard Gaussian, and $|\mathcal{N}(0, \mathrm{Id}_d)|$ (or $|\mathcal{N}_d|$ for short) to denote the positive standard half-Gaussian (i.e., $g \sim |\mathcal{N}(0, \mathrm{Id}_d)|$ if $g = |z|$ for $z \sim \mathcal{N}(0, \mathrm{Id}_d)$). We use $[n]$ to denote $\{1, \ldots, n\}$.

For $q > 0$, $\mathbb{Z}_q$ will denote the integers modulo q, which we will identify with $\{0, \ldots, q-1\}$. We use $\mathbb{Z}_q/q$ to denote $\{0, 1/q, \ldots, (q-1)/q\}$. Our discrete functions will in general have domain $\mathbb{Z}_q^d$ for some q. The $q = 2$ case, namely Boolean functions, have domain $\{0,1\}^d$. For the purposes of this chapter, sign : $\mathbb{R} \to \{0,1\}$ is defined as $\mathrm{sign}(t) = \mathbb{1}[t > 0]$. We will extend this to $\mathbb{Z}_q$ by defining $\mathrm{thres}_q : \mathbb{R} \to \mathbb{Z}_q$ in terms of a certain partition of $\mathbb{R}$ into q intervals $I_0, \ldots, I_{q-1}$ (formally defined later) as the piecewise constant function that takes on value k on I_k for each $k \in \mathbb{Z}_q$. Scalar functions and scalar arithmetic applied to vectors act elementwise. We say a quantity is *negligible* in a parameter n, denoted $\mathrm{negl}(n)$, if it decays as $1/n^{\omega(1)}$.

A *one-hidden-layer* ReLU network mapping $\mathbb{R}^d$ to $\mathbb{R}$ is a linear combination of ReLUs, that is, a function of the form

$$F(x) = W_1 \,\mathrm{ReLU}\left(W_0 x + b_0\right) + b_1, \tag{3.2.1}$$

where $W_0 \in \mathbb{R}^{k \times d}$, $W_1 \in \mathbb{R}^{1 \times k}$, $b_0 \in \mathbb{R}^k$, and $b_1 \in \mathbb{R}$. A *two-hidden-layer* ReLU network mapping $\mathbb{R}^d$ to $\mathbb{R}$ is a linear combination of ReLUs of one-hidden-layer networks, that is, a function of the form

$$F(x) = W_2 \,\mathrm{ReLU}\left(W_1 \,\mathrm{ReLU}\left(W_0 x + b_0\right) + b_1\right) + b_2, \tag{3.2.2}$$

where $W_0 \in \mathbb{R}^{k_0 \times d}$, $W_1 \in \mathbb{R}^{k_1 \times k_0}$, $W_2 \in \mathbb{R}^{1 \times k_1}$, $b_0 \in \mathbb{R}^{k_0}$, $b_1 \in \mathbb{R}^{k_1}$, and $b_2 \in \mathbb{R}$. Our usage of the term *hidden layer* thus corresponds to a *nonlinear* layer.

3.2.2 Learning models

Let $\mathcal{C}$ be a function class mapping $\mathbb{R}^d$ to $\mathbb{R}$, and let $\mathcal{D}$ be a distribution on $\mathbb{R}^d$. We consider various learning models where the learner is given access in different ways to labeled data $(x, f(x))$ for an unknown $f \in \mathcal{C}$ and must output a (possibly randomized) predictor that achieves (say) squared loss ϵ for any desired $\epsilon > 0$. In the traditional PAC model, access to the data is in the form of iid labeled examples $(x, f(x))$ where $x \sim \mathcal{D}$, and the learner is considered efficient if it succeeds using $\mathrm{poly}(d, 1/\epsilon)$ time and sample complexity. In the Statistical Query (SQ) model [Kea98, Rey20], access to the data is through an SQ oracle. Given a bounded query $\phi : \mathbb{R}^d \times \mathbb{R} \to [-1, 1]$ and a tolerance $\tau > 0$, the oracle may respond with any value v such that $|v - \mathbb{E}_{x \sim \mathcal{D}}[\phi(x, f(x))]| \le \tau$. A correlational query is one that is linear in y, i.e. of the form $\phi(x, y) = \widetilde{\phi}(x)y$ for some $\widetilde{\phi}$, and a correlational SQ (CSQ) learner is one that is only allowed to make CSQs. An SQ learner is considered efficient if it succeeds using $\mathrm{poly}(d, 1/\epsilon)$ queries and tolerance $\tau \ge 1/\mathrm{poly}(d, 1/\epsilon)$. Finally, in the label query model, the learner is allowed to request the value of $f(x)$ for any desired x, and is considered efficient if it succeeds using $\mathrm{poly}(d, 1/\epsilon)$ time and queries.

3.2.3 Learning with Rounding

The Learning with Rounding (LWR) problem [BPR12] is a close cousin of the well-known Learning with Errors (LWE) problem [Reg09], except with deterministic rounding in place of random additive errors.

Definition 3.2.1. Fix moduli $p, q \in \mathbb{N}$, where $p < q$, and let n be the security parameter. For any $w \in \mathbb{Z}_q^n$, define $f_w : \mathbb{Z}_q^n \to \mathbb{Z}_p/p$ by

$$f_w(x) = \frac{1}{p}\lfloor w \cdot x \rceil_p = \frac{1}{p}\left\lfloor \frac{p}{q}(w \cdot x \bmod q) \right\rceil,$$

where $\lfloor t \rceil$ is the closest integer to t. In the $\mathsf{LWR}_{n,p,q}$ problem, the secret w is drawn randomly from $\mathbb{Z}_q^n$, and we must distinguish between labeled examples (x, y) where $x \in \mathbb{Z}_q^n$ and either $y = f_w(x)$ or y is drawn independently from $\mathrm{Unif}(\mathbb{Z}_p/p)$. The $\mathsf{LWE}_{n,q,B}$ problem is similar, except that $y \in \mathbb{Z}_q/q$ is either $\frac{1}{q}((w \cdot x + e) \bmod q)$ for some $e \in \mathbb{Z}_q$ sampled from a carefully chosen distribution, e.g. discrete Gaussian, such that $|e| \le B$ except with $\mathrm{negl}(n)$ probability, or is drawn from $\mathrm{Unif}(\mathbb{Z}_q/q)$.

Remark 3.2.2. Traditionally the LWR problem is stated with labels lying in $\mathbb{Z}_p$ instead of $\mathbb{Z}_p/p$, although both are equivalent since the moduli p, q may be assumed to be known to the learner. The choice of $\mathbb{Z}_p/p$ is simply a convenient way to normalize labels to lie in $[0, 1]$. For consistency, we similarly normalize LWE labels to lie in $\mathbb{Z}_q/q$.

It is known that $\mathsf{LWE}_{n,q,B}$ is as hard as worst-case lattice problems when $q = \mathrm{poly}(n)$ and $B = q/\mathrm{poly}(n)$ (see e.g. [Reg10, Pei16] for surveys). Yet this is not known to directly imply the hardness of $\mathsf{LWR}_{n,p,q}$ in the regime in which p, q are both $\mathrm{poly}(n)$, which is the one we will be interested in as p, q will dictate the size of the hard networks that we construct in the proof of our cryptographic lower bound.

Unfortunately, in this polynomial modulus regime, it is only known how to reduce from LWE to LWR *when the number of samples is bounded relative to the modulus* [AKPW13, BGM+16]. For instance, the best known reduction in this regime obtains the following hardness guarantee:

Theorem 3.2.3 ([BGM+16])**.** *Let n be the security parameter, let $p, q \ge 1$ be moduli, and let $m, B \ge 0$. Assuming $q \ge \Omega(mBp)$, any distinguisher capable of solving $\mathsf{LWR}_{n,p,q}$ using m samples implies an efficient algorithm for $\mathsf{LWE}_{n,q,B}$.*

For our purposes, Theorem 3.2.3 is not enough to let us base our Theorem 3.1.2 off of LWE, as we are interested in the regime where the learner has an *arbitrary* polynomial number of samples.

LWR with polynomial modulus and arbitrary polynomial samples is nevertheless conjectured to be as hard as worst-case lattice problems [BPR12] and has already formed the basis for a number of post-quantum cryptographic proposals [DKRV18, CKLS18, BGML+18, JZ16]. We remark that one piece of evidence in favor of this conjecture is a reduction from a less standard variant of LWE in which the usual discrete Gaussian errors are replaced by errors uniformly sampled from the integers $f \quad q/2p, \ldots, q/2pg$ [BGM+16].

Note also that for our purposes we require *quasipolynomial*-time hardness (or $T(n)$-hardness for $T(n)$ being any other fixed, mildly superpolynomial function of the security parameter) of LWR. While slightly stronger than standard polynomial-time hardness, this remains a reasonable assumption since algorithms for worst-case lattice problems are still believed to require at least subexponential time.

3.2.4 Partial assignments

Let $\alpha \in f0, 1, \star g^d$ be a *partial assignment*. We refer to $S(\alpha) : fi \in [d] : \alpha_i = \star g \quad [d]$ as the set of *free variables* and $[d] n S(\alpha)$ as the set of *fixed variables*. Given two partial assignments α, β, let the *resolution* $\alpha \& \beta$ denote the partial assignment γ obtained by substituting α into β. That is,

$$\gamma_i = \begin{cases} \star & i \in S(\alpha) \setminus S(\beta) \\ \beta_i & i \in [d] n S(\beta) \\ \alpha_i & i \in S(\beta) n S(\alpha) \end{cases} \tag{3.2.3}$$

In this case we say that γ is a *refinement* of β that is the *result of applying* α. We write $\gamma \in \mathrm{App}(\alpha)$ to denote that γ is a result of applying α. Note that the set of refinements of β consists of all $3^{jS(\beta)j}$ partial assignments $\gamma \in f0, 1, \star g^d$ which agree with β on all fixed variables of β.

Given α, let $w(\alpha)$ denote $\{i : \alpha_i = 1\}$, that is, the Hamming weight of its fixed variables. Note that $w(\alpha \,\&\, \beta) \leq w(\alpha) + w(\beta)$.

Given a function $h : \mathbb{R}^d \to \mathbb{R}$ and partial assignment γ, we use $h_\gamma : \mathbb{R}^d \to \mathbb{R}$ to denote its partial restriction given by substituting in γ_i into the i-th input coordinate if $\gamma_i \in \{0, 1\}$. Note that given two partial restrictions α, β,

$$(h_\beta)_\alpha = h_{\alpha \,\&\, \beta} \tag{3.2.4}$$

We say that α is *sorted* if the restriction of α to its fixed variables is sorted in nonincreasing order, e.g. $\alpha = (1, \star, 1, \star, \star, 0, 0)$ is sorted, but $\alpha = (1, \star, 0, \star, \star, 0, 1)$ is not. Given α which is not necessarily sorted, denote its *sorting* by $\overline{\alpha}$. In general, we will use overline notation to denote sorted partial assignments.

3.3 Compressing the Daniely–Vardi Lift

In this section we show how to refine the lifting procedure of Daniely and Vardy [DV21] such that whenever the underlying discrete functions satisfy a property we term *compressibility*, we obtain hardness under the Gaussian for networks with just one extra hidden layer.

Definition 3.3.1. Let $q > 0$ be a modulus.[2] We call an L-hidden-layer ReLU network $f : \mathbb{Z}_q^d \to [0, 1]$ *compressible* if it is expressible in the form $f(x) = \sigma(h(x))$, where

- $h : \mathbb{Z}_q^d \to T$ is an $(L - 1)$-hidden-layer network such that $|h(x)| \leq \text{poly}(d)$ for all x;
- h has range $T = h(\mathbb{Z}_q^d)$ such that $T \subseteq \mathbb{Z}$ and $|T| \leq \text{poly}(d)$; and
- $\sigma : T \to [0, 1]$ is a mapping from h's possible output values to $[0, 1]$.

Remark 3.3.2. To see why such an f is an L-hidden-layer network in z, consider the function $\sigma : T \to \mathbb{R}$. Because $T \subseteq \mathbb{Z}$ and $|T| \leq \text{poly}(d)$, σ is expressible as (the

[2] Our results are stronger when q is taken to be a large polynomial in the dimension, but the Boolean $q = 2$ case is illustrative of all the main ideas.

restriction to T of) a piecewise linear function on $\mathbb{R}$ whose size and maximum slope are $\mathrm{poly}(d)$, and hence as a $\mathrm{poly}(d)$-sized one-hidden-layer ReLU network from $\mathbb{R}$ to $\mathbb{R}$. By composition, $x \mapsto \sigma(h(x))$ can be represented by an L-hidden-layer network.

We now formally state a theorem which captures our "compressed" version of the DV lift. The version of this theorem for $L+2$ layers is implicit in [DV21]. In technical terms, our improvement consists of removing the single outer ReLU present in their construction. Thus, while our construction still has three *linear* layers, it has only two *non-linear* layers.

Theorem 3.3.3 (Compressed DV lift)**.** *Let $q = \mathrm{poly}(d)$ be a modulus. Let $\mathcal{C}$ be a class of compressible L-hidden-layer $\mathrm{poly}(d)$-sized ReLU networks mapping $\mathbb{Z}_q^d$ to $[0,1]$. Let $m = m(d) = \omega_d(1)$ be a size parameter that grows slowly with d. There exists a class $\mathcal{C}^{\mathrm{M}}$ of $(L+1)$-hidden-layer $d^{(m)}$-sized ReLU networks mapping $\mathbb{R}^d$ to $[0,1]$ such that the following holds:*

Suppose there is an efficient algorithm A capable of learning $\mathcal{C}^{\mathrm{M}}$ over $N(0, \mathrm{Id}_d)$ up to squared loss $d^{-(m)}$. Then there is an efficient algorithm B capable of weakly predicting *$\mathcal{C}$ over $\mathrm{Unif}(\mathbb{Z}_q^d)$ with advantage $d^{-(m)}$ over guessing the constant $1/2$ in the following sense: given access to labeled examples $(x, f(x))$ for $x \sim \mathrm{Unif}(\mathbb{Z}_q^d)$ and an unknown $f \in \mathcal{C}$, B satisfies*

$$\mathbb{E}\left[\left(B(x) - f(x)\right)^2\right] < \mathbb{E}\left[\left(\frac{1}{2} - f(x)\right)^2\right] - d^{-(m)},$$

where the probability is taken over both x and the internal randomness of B. We refer to $\mathcal{C}^{\mathrm{M}}$ as the lifted class *corresponding to $\mathcal{C}$.*

By a standard padding argument, we obtain the following corollary which lets us work with polynomial-sized neural networks.

Corollary 3.3.4 (Compressed DV lift with padding)**.** *Let q, m and d be as above, and let $d' = d^m$. View $\mathcal{C}$ and $\mathcal{C}^{\mathrm{M}}$ as function classes on $\mathbb{Z}_q^{d'}$ and $\mathbb{R}^{d'}$ respectively, defined using only the first d coordinates, so that $\mathcal{C}^{\mathrm{M}}$ is now a $\mathrm{poly}(d')$-sized class over*

$\mathsf{R}^{d'}$. *Then an algorithm capable of learning* C^{M} *over* $\mathsf{N}_{d'}$ *up to squared loss* $1/\operatorname{poly}(d')$ *implies a weak predictor for* C *over* $\mathrm{Unif}(\mathsf{Z}_q^{d'})$ *with advantage* $1/\operatorname{poly}(d')$.

3.3.1 The DV Lift

Before proceeding to the proof of Theorem 3.3.3, we first outline the idea of the original DV lift in the setting of Boolean functions ($q = 2$). The goal is to approximate any given $f \in \mathsf{C}$ by a ReLU network $f^{\mathsf{DV}} : \mathsf{R}^d \to \mathsf{R}$ in such a way that f^{DV} under N_d behaves similarly to f under U_d. As a first attempt, one might consider the function $f^\star(z) = f(\mathrm{sign}(z))$ (also studied in [KK14a]), where recall that $\mathrm{sign}(t) = \mathbb{1}[t > 0]$. We could implement the following reduction: given a random example (x, y) where $x \sim U_d$ and $y = f(x)$, draw a fresh half-Gaussian $g \sim |\mathsf{N}_d|$ and output $((2x - 1)g, y)$ (where the arithmetic in defining the vector $(2x - 1)g$ is done elementwise). Since $2x - 1$ is distributed uniformly over $\{-1, 1\}^d$, the marginal is exactly N_d, and the labels are consistent with $f^\star$ since $\mathrm{sign}((2x - 1)g) = x$ and so $f(\mathrm{sign}((2x - 1)g)) = f(x)$. However, the issue is that the sign function is discontinuous, and so $f^\star$ is not realizable as a ReLU network.

Daniely and Vardi address this concern by devising a clever construction for f^{DV} that interpolates between two desiderata:

- For all but a small fraction of inputs, an initial layer successfully "Booleanizes" the input. In this case, one would like $f^{\mathsf{DV}}(z)$ to simply behave as $f(\mathrm{sign}(z))$.
- For the remaining fraction of inputs, we would ideally like f^{DV} to output an uninformative value such as zero, but this would violate continuity of f^{DV}.

The trick is to use a continuous approximation of the sign function, N_1, that interpolates linearly between 0 and 1 on an interval $[-\delta, \delta]$ (see Fig. 3.1a), and to pair it with a "soft indicator" function $N_2 : \mathsf{R} \to \mathsf{R}$ for the region where $N_1 \neq \mathrm{sign}$. Concretely, $N_2(t)$ is constructed as a one-hidden-layer ReLU network that (a) is always nonnegative, (b) equals 0 when $|t| \geq 2\delta$, and (c) equals 1 when $|t| \leq \delta$ (see Fig. 3.1b).

Now let $N_2^{\emptyset}(z) = \sum_j N_2(z_j)$, and define

$$f^{\mathsf{DV}}(z) = \mathrm{ReLU}(f(N_1(z)) \quad N_2^{\emptyset}(z)). \tag{3.3.1}$$

One can show that f^{DV} satisfies $f^{\mathsf{DV}}(z) = \mathrm{ReLU}(f(\mathrm{sign}(z)) \quad N_2^{\emptyset}(z))$, since $N_2^{\emptyset}$ "zeroes out" f^{DV} wherever N_1 6= sign for any coordinate. This lets us perform the following reduction: given examples (x, y) where $x \quad U_d$ and $y = f(x)$, draw a fresh $g \quad jN_dj$ and output $(z, \widetilde{y}) = ((2x \quad 1)g, \mathrm{ReLU}(y \quad N_2^{\emptyset}((2x \quad 1)g)))$. The marginal is again N_d, and the labels are easily seen to be consistent with f^{DV}. Correctness of the reduction can be established by using Gaussian anticoncentration to argue that f^{DV} is a good approximation of f. Formally, one can prove the following theorem.

Theorem 3.3.5 (Original DV lift, implicit in [DV21])**.** *Let C be a class of L-hidden-layer $\mathrm{poly}(d)$-sized ReLU networks mapping $f0, 1g^d$ to $[0, 1]$. There exists a class C^{DV} of $(L + 2)$-hidden-layer $\mathrm{poly}(d)$-sized ReLU networks mapping R^d to $[0, 1]$ such that the following holds. Suppose there is an e cient algorithm A capable of learning C^{DV} over $N(0, \mathrm{Id}_d)$ up to squared loss $\frac{1}{64}$. Then there is an e cient algorithm B capable of weakly predicting C over $\mathrm{Unif}f0, 1g^d$ with squared loss $\frac{1}{16}$.*

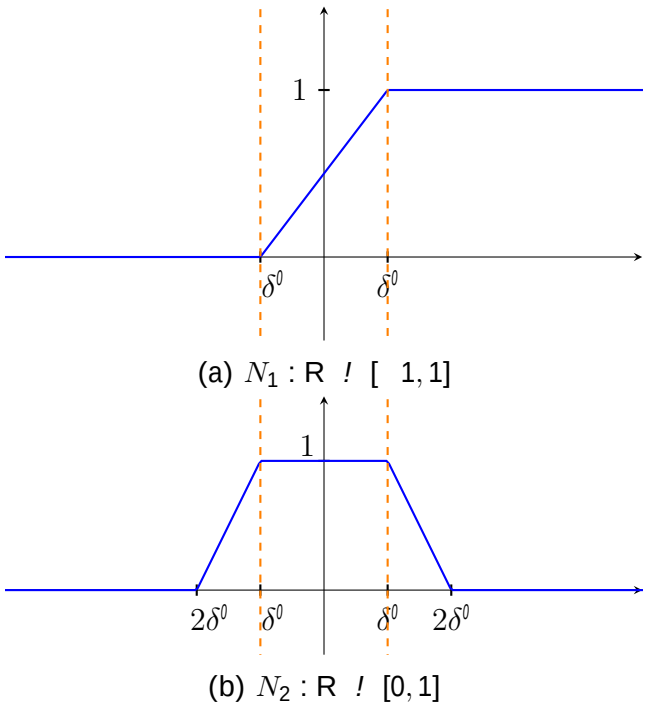


Figure 3.1: Schematic plots of N_1 and N_2 in the $q = 2$ case, where $N_2^0(z)$ may be realized as $\sum_{j2[d]} N_2(z_j)$. Here, $\delta^0 = \Theta(\delta)$ where δ is the parameter from Lemmas 3.3.6 and 3.3.7.

We now show how to construct the gadgets N_1 and N_2, extending them to make them suitable for working with Z_q for general q as opposed to just $\mathit{f}0, 1\mathit{g}$. These constructions utilize the simple but important property that piecewise linear functions on the real line are readily and efficently realized as linear combination of ReLUs.

Start by letting $I_0, I_1, \ldots, I_{q\;\;1}$ be a partition of R into q consecutive intervals each of mass $1/q$ under $\mathit{N}(0, 1)$ (e.g., when $q = 2$, $I_0 = (\;\;\mathit{1}, 0)$ and $I_1 = (0, \mathit{1}\;)$). Note that these intervals will have differing lengths, and the shortest ones will be the ones closest to the origin. Still, by Gaussian anti-concentration, we know that each $\mathit{j}I_j\mathit{j}\;\;\;\Theta(1/q)$. Let $\text{thres}_q : \mathsf{R} \;\; ! \;\; \mathsf{Z}_q$ be the piecewise constant function that takes on value k on I_k. Clearly, when $t \;\;\; \mathit{N}(0, 1)$, $\text{thres}_q(t) \;\;\; \text{Unif}(\mathsf{Z}_q)$. Let $R_1, \ldots, R_q$ be intervals such that $R_k \;\;\; I_{k\;\;1} \; \mathit{[}\; I_k$ and R_k contains the boundary point between $I_{k\;\;1}$

and I_k, and such that each R_k has mass δ/q for some δ 1 to be picked later. Let $S_1, \ldots, S_q$ be slightly larger intervals such that R_k S_k for each k 2 $[q$ 1], and each S_k has mass $2\delta/q$. By Gaussian anti-concentration again, each jS_kj $\Theta(\delta/q)$. Notice that by construction, $\mathsf{P}_{z\ N(0,1)}[z$ 2 $[_k R_k] = \delta$ and $\mathsf{P}_{z\ N(0,1)}[z$ 2 $[_k S_k] = 2\delta$.

Lemma 3.3.6. *Let $\delta > 0$, $q > 0$, and intervals I_k, R_k, S_k for k 2 Z_q be as above. There exists a one-hidden-layer ReLU network $N_1 : \mathsf{R}$! R with $O(q)$ units and weights of magnitude $O(q/\delta)$ such that $N_1(t) = \mathrm{thres}_q(t)$ if t 2 $[_k R_k$.*

Proof. This can be done by considering the piecewise linear function that approximates the function thres_q by matching it exactly on R n $[_k R_k$, and interpolating linearly between values k 1 and k on the interval R_k for each k 2 $[q$ 1]. □

Lemma 3.3.7. *Let $\delta > 0$, $q > 0$, and intervals I_k, R_k, S_k for k 2 Z_q be as above. There exists a one-hidden-layer ReLU network $N_2 : \mathsf{R}$! $[0,1]$ with $O(q)$ units and weights of magnitude $O(q/\delta)$ such that*

$$N_2(t) \text{ is } \begin{cases} = 1 & \text{if } t \text{ 2 } [_k R_k \\ = 0 & \text{if } t \text{ 2 } \mathsf{R} \text{ n } [_k S_k \\ 0 & \text{otherwise} \end{cases}.$$

Proof. Consider the piecewise linear function that is 0 on Rn $[_k S_k$, is 1 on $[_k R_k$, and interpolates linearly between 0 and 1 (or 1 and 0) on S_k n R_k for every k 2 $[q$ 1]. Put differently, the graph of N_2 consists of a trapezoid on each S_k that achieves its maximum value of 1 on R_k. □

3.3.2 Saving One Hidden Layer via Compressibility

The starting point for exploiting compressibility to avoid a hidden layer in the lift is as follows. Compressibility lets us express $f(x)$ as $\sigma(h(x))$ for some $h : \mathsf{Z}_q^d$! T with a poly(d)-sized range T Z, and some $\sigma : T$! $[0,1]$. So we can write

$$f(x) = \sigma(h(x)) = \sum_{t\ 2T} \sigma(t\ )\mathbb{1}[h(x) = t\ ].$$

We would like a lifted function $f^{\mathsf{M}} : \mathsf{R}^d \; ! \; \mathsf{R}$ (where we introduce f^{M} as notation to distinguish our lift from the original DV lift, denoted f^{DV}) such that $f^{\mathsf{M}}(z)$ behaves like $\sigma(h(\text{thres}_q(z)))$ except when N_2 indicates that $N_1 \; 6= \; \text{thres}_q$, in which case we want $f^{\mathsf{M}}(z) = 0$. To this end, we might hope to write

$$f^{\mathsf{M}}(z) = \sum_{t \; 2T} \sigma(t \;) \mathbb{1}[h(N_1(z)) = t \;] \mathbb{1}[8j : N_2(z_j) < 1].$$

Suppose that we could build a one-hidden-layer network $N_3(s_1, \ldots, s_d; t)$ that behaves like $\mathbb{1}[t = 0]\mathbb{1}[8j : s_j < 1]$. Then we could realize f^{M} as

$$f^{\mathsf{M}}(z) = \sum_{t \; 2T} \sigma(t \;) N_3(N_2(z_1), \ldots, N_2(z_d); \; h(N_1(z)) \quad t \;).$$

Notice that whenever $N_2(z_j) = 1$ for any coordinate j, this expression vanishes. Otherwise, we know that $h(N_1(z)) = h(\text{thres}_q(z))$, which takes values in T, so that only the summand with $t \;\; = h(\text{thres}_q(z))$ survives and the expression simplifies to $f(\text{thres}_q(z))N_3(N_2(z_1), \ldots, N_2(z_d); 0)$. It is not hard to show that this is sufficient to let us complete the required reduction. Moreover, because N_3 is a one-hidden-layer network in its arguments, and because both $h \quad N_1$ and N_2 have at most L hidden layers (for $h \quad N_1$, one comes from N_1 and $L \quad 1$ from h; for N_2, it itself has just one hidden layer), this implementation of f^{M} would have only $L + 1$ hidden layers.

Slightly more generally, one can show that it would suffice to build a one-hidden-layer network N_3 with the following properties:

$$N_3(s_1, \ldots, s_d; t) = \begin{cases} 0 & \text{if } 9j : s_j = 1 \\ 0 & \text{if } t \; 2 \; \mathsf{Z} \; n \; f0g \\ 1 & \text{if } 8j : s_j = 0 \text{ and } t = 0 \end{cases} \tag{3.3.2}$$

Unfortunately, most natural attempts to construct N_3 with such ideal properties — in particular, all formulations of N_3 purely as a function of two variables, $\sum_j s_j$ and t, which was the approach taken in [DV21] — run into difficulties and appear to require *two* hidden layers (see Appendix B.1 for discussion). One approach that does almost work, however, comes at the cost of exponential size. Let $\psi(s_1, \ldots, s_d; t)$

be any function that vanishes whenever $t \in \mathbb{Z} \setminus \{0\}$ (for all $s_1, \ldots, s_d \in [0,1]^d$). For simplicity, let us consider the $d = 3$ case. Consider the following expression that resembles the inclusion-exclusion formula:

$$\psi(s_1, s_2, s_3; t) \quad \psi(1, s_2, s_3; t) \quad \psi(s_1, 1, s_3; t) \quad \psi(s_1, s_2, 1; t) \tag{3.3.3}$$

$$+ \psi(s_1, 1, 1; t) + \psi(1, s_2, 1; t) + \psi(s_1, 1, 1; t) \quad \psi(1, 1, 1; t) \tag{3.3.4}$$

Notice that whenever any $s_j = 1$, this expression vanishes identically. Moreover, for any $t \in \mathbb{Z} \setminus \{0\}$ (and any $s_1, \ldots, s_d$), the expression vanishes again because each summand vanishes. Thus the first two properties are satisfied; the third property turns out to be more subtle, and we will ignore it for the moment. The natural generalization of this expression to general d can be stated in the language of partial assignments.

Lemma 3.3.8. *Let $\psi : \mathbb{R}^d \to \mathbb{R}$ be any function. Let P_i denote the set of partial assignments $\gamma \in \{1, \star\}^d$ with i 1s. The expression*

$$\sum_{i=0}^{d} \sum_{\gamma \in P_i} (-1)^i \psi_\gamma \tag{3.3.5}$$

vanishes whenever any $s_j = 1$. (We may view t as an additional parameter that is always left free, as in Eq. (3.3.3))

Proof. For concreteness, suppose $s_1 = 1$. Let $P_i^\star$ (resp. P_i^1) denote the set of $\gamma \in P_i$ with $s_1 = \star$ (resp. $s_1 = 1$). For every $i \in \{0, \ldots, d-1\}$, we can form a bijection between $P_i^\star$ and P_{i+1}^1 using the map $\gamma \mapsto \gamma^\emptyset$ where $\gamma^\emptyset = (1, \gamma_2, \ldots, \gamma_d)$. When $s_1 = 1$, for every such pair $(\gamma, \gamma^\emptyset)$, we have $\psi_\gamma = \psi_{\gamma^\emptyset}$, and moreover they occur in (3.3.5) with opposite signs. Thus the entire expression vanishes. □

Let us assume for now that ψ is picked suitably and the rest of the reduction goes through with this construction (as one can verify when we come to the proof of Theorem 3.3.3, this would indeed be the case). This construction has size 2^d,

meaning that the resulting lifted functions would have size $S = \text{poly}(2^d)$. But by Theorem 3.4.5, the SQ lower bound for the LWR functions over $\mathbb{Z}_q^n$ with $n = d$ and $q = \text{poly}(n)$ scales as $q^{\Omega(n)} = 2^{\Omega(d \log d)} = S^{\Omega(\log \log S)}$, which is still superpolynomial in S. Thus after padding the dimension to $d' = 2^d$, this construction would actually still yield a superpolynomial SQ lower bound for two-hidden-layer ReLU networks over $\mathbb{R}^{d'}$.

Instead of pursuing this route, however, we give a more efficient construction that has size only slightly superpolynomial in d. The key idea is to restrict attention to those possibilities for $(s_1, \ldots, s_d) = (N_2(z_1), \ldots, N_2(z_d))$ that are the most likely. Specifically, if $m = \omega_d(1)$ is the size parameter from Theorem 3.3.3, then by setting δ in Lemmas 3.3.6 and 3.3.7 appropriately, we can ensure that with overwhelming probability over $z \sim \mathcal{N}(0, \text{Id})$, no more than m of the $N_2(z_j)$ are simultaneously 1. Accordingly, we focus on constructing N_3 such that

$$N_3(s_1, \ldots, s_d; t) = \begin{cases} 0 & \text{if between 1 and } m \text{ of the } s_i \text{ are 1} \\ 0 & \text{if } t \in \mathbb{Z} \setminus \{0\} \\ 1 & \text{otherwise} \end{cases}. \tag{3.3.6}$$

We now describe a $d^{O(m)}$-sized construction for N_3 that satisfies the first and second properties exactly, and "approximately" satisfies the third in the sense that it takes on a nonzero value with nonnegligible probability over its inputs. As we will see later, this turns out to be enough for the reduction to go through. The construction retains the spirit of using a linear combination of partial restrictions.

Lemma 3.3.9 (Main lemma)**.** *Let $m = m(d) = \omega_d(1)$ be a size parameter. Let A denote the set of all partial assignments $\alpha \in \{0, 1, \star\}^d$ for which $|S(\alpha)| = m$ and $w(\alpha) = 1$. Let B denote the set of all sorted partial assignments given by refining some element of A and sorting. Given $i, j \geq 0$, let $B_{i,j}$ denote the set of $\overline{\beta} \in B$ for which $|S(\overline{\beta})| = i$ and $w(\overline{\beta}) = j$. For any symmetric function $\psi : \mathbb{R}^d \to \mathbb{R}$, define the*

function

$$\psi \quad , \ \psi \quad \sum_{i=0}^{m} \sum_{j=1}^{m+1} (\quad 1)^{m \ i} \ \lambda_{i+j} \sum_{\overline{\beta} 2 B_{i,j}} \psi_{\overline{\beta}}, \qquad \textit{for } \lambda_k \ , \ \begin{pmatrix} d & k & 1 \\ m & k+1 \end{pmatrix} \tag{3.3.7}$$

Then

(a) *jBj* $\binom{d}{m}(d \quad m) \ 3^m$

(b) ψ *is symmetric*

(c) $\psi_\alpha : \mathsf{R}^d$ *!* R *is the identically zero function for all* α *2 A.*

Lemma 3.3.10. *Let*

$$\psi(s_1, \ldots, s_d; t) = \sum_{i=1}^{d} \mathrm{ReLU}\left(t \quad \left(s_i \quad \frac{1}{d \quad 1} \sum_{j \in i} s_j \right) \right) \quad \mathrm{ReLU}(dt),$$

viewed as a function of $s_1, \ldots, s_d$ *parameterized by* t*, and let* ψ *be as above. De ne* $N_3(s_1, \ldots, s_d; t) = \psi \ (s_1, \ldots, s_d; t)$*. Then*

(a) $N_3(s_1, \ldots, s_d; t) = 0$ *for any* t *2* R *if between 1 and* m *of the* s_j *are 0*

(b) $N_3(s_1, \ldots, s_d; t) = 0$ *for any* $s_1, \ldots, s_d$ *2* $[0, 1]^d$ *if* t *2* Z *n f*0*g*

(c) N_3 *has size at most* d^{2m}

(d) $N_3(\underbrace{0, \ldots, 0}_{d \ 1}, s; 0) = s$ *for any* s *2* $[0, \frac{1}{d}]$.

Before proceeding to the proofs of Lemmas 3.3.9 and 3.3.10, let us see how to use them to prove Theorem 3.3.3.

Proof of Theorem 3.3.3. For each f *2* C given by $f = \sigma \quad h$, let f^{M} *2* C^{M} be given by

$$f^{\mathsf{M}}(z) = \sum_{t \ 2T} \sigma(t \ ) N_3(N_2(z_1), \ldots, N_2(z_d); \ h(N_1(z)) \quad t \ ), \tag{3.3.8}$$

where N_1 and N_2 are from Lemmas 3.3.6 and 3.3.7, with the δ parameter set to d^{-10m}, and N_3 is from Lemma 3.3.10. This is an $(L+1)$-hidden layer network since h N_1 and N_2 each have at most L hidden layers, and N_3 adds an additional layer. By Lemma 3.3.10(c), the size of this network is $S = d^{(m)}$. Note that whenever z is such that $N_2(z_1), \ldots, N_2(z_d) < 1$, then:

- $N_1(z) = \text{thres}_q(z)$, and so $h(N_1(z)) = h(\text{thres}_q(z))$ takes only integer values in $T = h(\mathsf{Z}_q^d)$; and
- the only t for which one of the summands in Eq. (3.3.8) is potentially nonzero is the one given by $t = h(\text{thres}_q(z))$.

Thus in this case f^{M} simplifies to

$$f^{\mathsf{M}}(z) = \sigma(h(\text{thres}_q(z)))\ N_3(N_2(z_1), \ldots, N_2(z_d);\ 0) \tag{3.3.9}$$

$$= f(\text{thres}_q(z))\ N_3(N_2(z_1), \ldots, N_2(z_d);\ 0). \tag{3.3.10}$$

Further, for z such that between 1 and m of the $N_2(z_j)$ are 1, we know that $\psi(N_2(z_1), \ldots, N_2(z_d);\ t) = 0$ identically (for all t 2 R), so in this case $f^{\mathsf{M}}(z) = 0$. And finally, for z such that more than m of the $N_2(z_j)$ are 1, we have no guarantees on the behavior of f^{M}, but as we now show, we have set parameters such that this case occurs only with negligible probability, and we can pretend that 0 is still a valid label in this case. Indeed, by standard Gaussian anti-concentration, for each coordinate z_j we have $\mathsf{P}_{z_j}[N_2(z_j) = 1] = \mathsf{P}_{z_j}[z_j\ 2\ [_k R_k] = \delta = d^{-10m}$. The number of coordinates j for which $N_2(z_j) = 1$ thus follows a binomial distribution $B(d, d^{-10m})$, which has a decreasing pdf with unique mode at $\mathsf{b}(d+1)d^{-10m}\mathsf{c} = 0$. Thus the probability of having at least m 1s is at most

$$\sum_{i=m}^{d} \binom{d}{i} (d^{-10m})^i (1 \quad d^{-10m})^{d-i} \quad (d \quad m+1)\binom{d}{m} d^{-10m^2} \quad dd^m d^{-10m^2} \quad d^{-9m^2} \tag{3.3.11}$$

for sufficiently large d. This is negligibly small not only in d but in the size of the network, $S = d^{(m)}$.

We now describe the reduction. For each labeled example (x, y) that the discrete learner B receives, where x Unif(Z_q^d) and $y = f(x)$ for an unknown f 2 C, B forms a labeled example $(z, \widetilde{y})$ for the Gaussian learner A as follows. For each coordinate j 2 $[d]$, z_j is drawn from $\mathsf{N}(0, 1)$ conditioned on z_j 2 I_{x_j}. Notice that this way $\text{thres}_q(z) = x$, and the marginal distribution on z is exactly N_d. The modified label is given by

$$\widetilde{y} = \widetilde{y}(y, z) = \begin{cases} 0 & \text{if more than } m \text{ of the } N_2(z_j) \text{ are } 1 \\ 0 & \text{if between 1 and } m \text{ of the } N_2(z_j) \text{ are } 1 \\ y\ N_3(N_2(z_1), \ldots, N_2(z_d);\ 0) & \text{otherwise} \end{cases} \tag{3.3.12}$$

Note that in the bottom two cases, $\widetilde{y} = f^{\mathsf{M}}(z)$ exactly; in the top case $\widetilde{y}$ is in general inconsistent with f^{M}, but as we have seen, this case occurs with $\text{negl}(S)$ probability. In particular, with overwhelming probability, no $\text{poly}(S)$-time algorithm will ever see non-realizable samples.

So B can feed these new labeled examples $(z, \widetilde{y})$ to A. Suppose A outputs a hypothesis $\widehat{f} : \mathsf{R}^d$! R such that $\mathsf{E}_{z\ N_d}[(\widehat{f}(z)\ f^{\mathsf{M}}(z))^2]$ ϵ. We need to show B can convert this hypothesis into a nontrivial one for its discrete problem. We first define a "good region" G R^d where f^{M} is guaranteed to be nonzero and nontrivially related to the original f by saying z 2 G iff $N_2(z_1), \ldots, N_2(z_{d\ 1}) = 0$, and $N_2(z_d)$ 2 $(\frac{1}{2d}, \frac{1}{d})$. Observe that when z 2 G, by Eq. (3.3.10) and Lemma 3.3.10(d) we have

$$f^{\mathsf{M}}(z) = f(\text{thres}_q(z))N_3(N_2(z_1), \ldots, N_2(z_{d\ 1}), N_2(z_d); 0) \tag{3.3.13}$$

$$= f(x)N_3(0, \ldots, 0, N_2(z_d); 0) \tag{3.3.14}$$

$$= yN_2(z_d), \tag{3.3.15}$$

where we use the fact that $\text{thres}_q(z) = x$, so that $f(\text{thres}_q(z)) = f(x) = y$. Let us compute the probability mass of G. For coordinates j 2 $[d$ $1]$, note that $\mathsf{P}[N_2(z_j) = 0] = \mathsf{P}[z_j$ 2 $[_k S_k] = 1$ $2\delta = 1$ d $^{(m)}$. For z_d, we need a lower bound on the probability that $N_2(z_d)$ 2 $(\frac{1}{2d}, \frac{1}{d})$. Consider the behavior of N_2 on just the interval S_k that is closest to the origin (which will be $k = \mathsf{d}q/2\mathsf{e}$): it changes linearly from

0 to 1 (and again from 1 to 0) on $S_k \cap R_k$. It is not hard to see that N_2 takes values in $(\frac{1}{2d}, \frac{1}{d})$ on a $O(1/d)$ fraction of S_k. Since the Gaussian pdf will be at least some constant on all of S_k, the probability that z_d lands in this fraction of S_k is $\Omega(|S_k|/d) = \Omega(\delta/qd)$ $d^{(m)}$. Overall, we get that

$$\mathbb{P}[z \in R] = \mathbb{P}\left[N_2(z_d) \in \left(\frac{1}{2d}, \frac{1}{d}\right)\right] \prod_{j \in [d-1]} \mathbb{P}[N_2(z_j) = 0] \quad (1 - d^{(m)})^{d-1} d^{(m)} = d^{(m)},$$

which is still $1/\operatorname{poly}(S)$ and hence non-negligible in the size S of the network.

The discrete learner B can now adapt $\widehat{f}$ as follows. Given a fresh test point $x \sim \mathrm{Unif}(\mathbb{Z}_q^d)$, the learner forms z such that for each coordinate $j \in [d]$, z_j is drawn from $\mathcal{N}(0,1)$ conditioned on $z_j \in I_{x_k}$; for brevity, we shall denote the random variable z conditioned on x (formed in this way) by $z|x$. If $z \in G$, then B predicts $\widehat{y} = \frac{\widehat{f}(z)}{N_2(z_d)}$ (recall that when $z \in z$, $N_2(z_d) > \frac{1}{2d}$), and otherwise it simply predicts $\widetilde{y} = \frac{1}{2}$. The

square loss of this predictor is given by

$$\mathop{\mathbb{E}}_{x \sim \mathrm{Unif}(\mathbb{Z}_q^d)}[(\widehat{y} - f(x))^2] = \mathop{\mathbb{E}}_{x}\mathop{\mathbb{E}}_{z|x}[(\widehat{y} - f(x))^2] \tag{3.3.16}$$

$$= \mathop{\mathbb{E}}_{x,z|x}[(\widehat{y} - f(x))^2 \mid z \in G]\,\mathbb{P}[z \in G] + \mathop{\mathbb{E}}_{x,z|x}[(\widehat{y} - f(x))^2 \mid z \notin G]\,\mathbb{P}[z \notin G] \tag{3.3.17}$$

$$= \mathop{\mathbb{E}}_{x,z|x}\left[\left(\frac{\widehat{f}(z)}{N_2(z_d)} - f(x)\right)^2 \mid z \in G\right]\mathbb{P}[z \in G] + \mathop{\mathbb{E}}_{x,z|x}\left[\left(\frac{1}{2} - f(x)\right)^2 \mid z \notin G\right]\mathbb{P}[z \notin G] \tag{3.3.18}$$

$$= \mathop{\mathbb{E}}_{x,z|x}\left[\left(\frac{\widehat{f}(z)}{N_2(z_d)} - \frac{f^{\mathsf{M}}(z)}{N_2(z_d)}\right)^2 \mid z \in G\right]\mathbb{P}[z \in G] + \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right]\mathbb{P}[z \notin G]$$

(by Eq. (3.3.15), when $z \in G$, $f^{\mathsf{M}}(z) = f(x)N_2(z_d)$)

$$< 4d^2\,\mathop{\mathbb{E}}_{z}[(\widehat{f}(z) - f^{\mathsf{M}}(z))^2 \mid z \in G]\,\mathbb{P}[z \in G] + \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right]\mathbb{P}[z \notin G]$$

(when $z \in G$, $N_2(z_d) > \frac{1}{2d}$)

$$4d^2\,\mathop{\mathbb{E}}_{z}[(\widehat{f}(z) - f^{\mathsf{M}}(z))^2] + \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right]\mathbb{P}[z \notin G] \tag{3.3.19}$$

$$4d^2\epsilon + \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right]\mathbb{P}[z \notin G] \tag{3.3.20}$$

$$= \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right] + 4d^2\epsilon - \mathop{\mathbb{E}}_{x}\left[\left(\frac{1}{2} - f(x)\right)^2\right]\mathbb{P}[z \in G]. \tag{3.3.21}$$

In the case of the hard classes C that we consider, we may assume without loss of generality that $\mathbb{E}_{x \sim \mathrm{Unif}(\mathbb{Z}_q^d)}[(\frac{1}{2} - f(x))^2] \geq 1/\mathrm{poly}(d)$, since otherwise the problem of learning C is trivial (in fact, in our applications we will have $\mathbb{E}_{x \sim \mathrm{Unif}(\mathbb{Z}_q^d)}[(\frac{1}{2} - f(x))^2] = \Theta(1)$). This means that by taking

$$\epsilon = \mathbb{P}[z \in G]/\mathrm{poly}(d) = d^{-(m)}/\mathrm{poly}(d) = d^{-(m)}$$

sufficiently small (but still $1/\mathrm{poly}(S)$), we may ensure that the square loss of the discrete learner B is at most $\mathbb{E}_{x \sim \mathrm{Unif}(\mathbb{Z}_q^d)}[(\frac{1}{2} - f(x))^2] - d^{-(m)}$, as desired. □

Remark 3.3.11. The only property of the Gaussian $\mathcal{N}(0, \mathrm{Id}_d)$ used crucially in the proof above is that it is a product distribution $P = \bigotimes_{i \in [d]} P_i$ where each P_i is suitably anti-concentrated. By some simple changes to the parameters of N_1, N_2 and N_3

(depending on P), the proof can be made to work more generally for such distributions P.

3.3.3 Proofs of Lemmas 3.3.9 and 3.3.10

We now detail the proofs involved in the construction of the gadget N_3.

Proof of Lemma 3.3.9. Note that $|A| = \binom{d}{m}(d \quad m)$. Any partial assignment β has at most $3^{|S(\beta)|}$ refinements, and B is a subset of all refinements of partial assignments from A, so $|B| \quad \binom{d}{m}(d \quad m) \quad 3^m$.

For the remaining parts of the lemma, it will be useful to observe that B consists exactly of all partial assignments with i free variables and j 1s for any 0 $i \quad m$ and $j \quad 1$ satisfying $i + j \quad m + 1$.

To prove the second part of the lemma, it suffices to show that

$$\sum_{\overline{\beta} 2 B_{i,j}} h_{\overline{\beta}} \tag{3.3.22}$$

is symmetric for all i, j. As transpositions generate the symmetric group on d elements, it suffices to show that (3.3.22) is invariant under swapping two input coordinates, call them $a, b \; 2 \; [d]$. For all $\overline{\beta} \; 2 \; B_{i,j}$ for which a, b are either both present or both absent in $S(\overline{\beta})$, this clearly does not affect the value of $h_{\overline{\beta}}$. Now consider the set S_a (resp. S_b) of partial assignments $\overline{\beta} \; 2 \; B_{i,j}$ for which only a (resp. only b) is present in $S(\overline{\beta})$. There is a clear bijection $f : S_a \; ! \; S_b$: given $\overline{\beta} \; 2 \; S_a$, swap the a- and b-th entries, and vice-versa, and for any $\overline{\beta} \; 2 \; S_a$, the function $h_{\overline{\beta}} + h_{f(\overline{\beta})}$ is unaffected by the swapping of input coordinates a, b. This concludes the proof of the second part of the lemma.

Finally, to prove the third part of the lemma, it suffices to verify it for a single $\alpha \; 2 \; A$, as h is symmetric. So consider $\overline{\alpha} = f1, 0, \quad , 0, \star, \quad , \star g$. We apply (3.2.4)

to get

$$h_{\overline{\alpha}} = h_{\overline{\alpha}} \sum_{i=0}^{m} \sum_{j=1}^{m+1} (-1)^{m-i} \lambda_{i+j} \sum_{\overline{\beta} \in B_{i,j}} h_{\overline{\alpha} \,\&\, \overline{\beta}} \tag{3.3.23}$$

$$= h_{\overline{\alpha}} \sum_{\overline{\gamma} \in B \setminus \mathrm{App}(\alpha) \text{ sorted}} h_{\overline{\gamma}} \sum_{i=0}^{m} \sum_{j=1}^{m+1} (-1)^{m-i} \lambda_{i+j} \sum_{\overline{\beta} \in B_{i,j}} \mathbb{1}[\overline{\overline{\alpha} \,\&\, \overline{\beta}} = \overline{\gamma}] \tag{3.3.24}$$

Note that for $\overline{\gamma} = \overline{\alpha}$, the only $\overline{\beta} \in B$ for which $\overline{\overline{\alpha} \,\&\, \overline{\beta}} = \overline{\gamma}$ is $\overline{\beta} = \overline{\alpha}$. Indeed, for $\overline{\beta}$ to be such that $\overline{\overline{\alpha} \,\&\, \overline{\beta}} = \overline{\alpha}$, it must have $S(\overline{\beta}) = S(\overline{\alpha})$ and exactly one 1, from which it follows that $\overline{\beta} = \overline{\alpha}$. Since $\overline{\alpha} \in B_{m,1}$, its coefficient in (3.3.24) is given by

$$(-1)^{m-m} \lambda_{m+1} = 1, \tag{3.3.25}$$

and so the h_α in (3.3.24) cancels with the $\overline{\gamma} = \overline{\alpha}$-th summand in (3.3.24).

In the rest of the proof, we can thus focus on sorted $\overline{\gamma} \in B \setminus \mathrm{App}(\alpha) \{\overline{\alpha}\}$. Note that such $\overline{\gamma}$ satisfy

$$|S(\overline{\gamma})| < m. \tag{3.3.26}$$

To see this, recall that any $\overline{\gamma} \in B$ with $|S(\overline{\gamma})| = m$ must have exactly one 1, and since $\overline{\gamma} \in \mathrm{App}(\overline{\alpha})$ it must be that $\overline{\gamma}$ must have $S(\overline{\gamma}) = S(\overline{\alpha})$ and so $\overline{\gamma} = \overline{\alpha}$.

Observe that we must have $\overline{\gamma}_1 = 1$. Indeed, it cannot be 0 because $\overline{\gamma}$ is sorted and has at least one 1. It also cannot be $\star$. To see this, consider any $\overline{\beta}$ for which $\overline{\overline{\alpha} \,\&\, \overline{\beta}} = \overline{\gamma}$. If we had $\overline{\beta}_1 \neq \star$, then clearly $\overline{\gamma}_1 \neq \star$. If we had $\overline{\beta}_1 = \star$, then $(\overline{\alpha} \,\&\, \overline{\beta})_1 = 1$ (as $\overline{\alpha}_1 = 1$), so $\overline{\gamma} = \overline{\overline{\alpha} \,\&\, \overline{\beta}}$ must also have first entry given by 1.

We are now ready to calculate the coefficient of $h_{\overline{\gamma}}$ (for each $\overline{\gamma} \in B \setminus \mathrm{App}(\alpha) \{\overline{\alpha}\}$) in (3.3.24) by adding the coefficients of all the $\overline{\beta} \in B$ for which

$$\overline{\overline{\alpha} \,\&\, \overline{\beta}} = \overline{\gamma}. \tag{3.3.27}$$

First let us consider the contribution of $\overline{\beta} \in B$ for which $\overline{\beta}_1 = 1$. Observe that such $\overline{\beta}$ must have exactly $w(\overline{\gamma})$ 1s. Furthermore, such a $\overline{\beta}$ is an element of B if and

only if it has at most $m+1 \quad w(\overline{\gamma})$ free variables, and the set of free variables in $\overline{\beta}$ must be $S(\overline{\gamma}) \ [\ V$ where V is any subset of $[d] \n (\{1\} \ [\ S(\overline{\alpha}))$. The contribution of all such $\overline{\beta}$ to the coefficient of $h_{\overline{\gamma}}$ in (3.3.24) is thus

$$\sum_{i=jS(\overline{\gamma})j}^{m+1 \quad w(\overline{\gamma})} (\quad 1)^{m \quad i} \ \lambda_{i+w(\overline{\gamma})} \ \begin{pmatrix} d & m & 1 \\ i & jS(\overline{\gamma})j \end{pmatrix}, \tag{3.3.28}$$

where here the index i denotes the total number of free variables in $\overline{\beta}$, and the factor of $\binom{d \ m \ 1}{i \ jS(\overline{\gamma})j}$ is the number of ways to choose V.

It remains to consider the contribution from $\overline{\beta} \ 2 \ B$ for which $\overline{\beta}_1 \ 6= 1$. First note that clearly we cannot have $\overline{\beta}_1 = 0$, as $\overline{\beta}$ is sorted and has at least one 1 because it lies in B. The only possibility is $\overline{\beta}_1 = \star$, which we split into two cases based on $w(\overline{\gamma})$.

Case 1: $w(\overline{\gamma}) = 1$**.** In this case, we claim that there are no $\overline{\beta} \ 2 \ B$ simultaneously satisfying (3.3.27) and $\overline{\beta}_1 = \star$. Suppose to the contrary. Then such a $\overline{\beta}_1$ must have at least one 1 in some other entry (as $\overline{\beta} \ 2 \ B$), but this would imply that the resolution $\overline{\alpha} \ \& \ \overline{\beta}$ has at least two 1s, a contradiction. The total coefficient of $h_{\overline{\gamma}}$ in this case is thus exactly given by (3.3.28). Upon substituting $w(\overline{\gamma}) = 1$, this simplifies to

$$\sum_{i=jS(\overline{\gamma})j}^{m+1 \quad w(\overline{\gamma})} (\quad 1)^{m \quad i} \ \lambda_{i+1} \begin{pmatrix} d & m & 1 \\ i & jS(\overline{\gamma})j \end{pmatrix} = \sum_{i=jS(\overline{\gamma})j}^{m+1 \quad w(\overline{\gamma})} (\quad 1)^{m \quad i} \begin{pmatrix} d & i & 2 \\ d & m & 2 \end{pmatrix} \begin{pmatrix} d & m & 1 \\ i & jS(\overline{\gamma})j \end{pmatrix} = 0, \tag{3.3.29}$$

where in the last step we use Lemma B.2.1 (which we can apply because of (3.3.26)).

Case 2: $w(\overline{\gamma}) > 1$**.** Observe that we must have $w(\overline{\beta}) = w(\overline{\gamma}) \quad 1$ (as the only entry of $\overline{\alpha}$ equal to 1 is the first entry, and the first entry of $\overline{\beta}$ is $\star$). As $w(\overline{\gamma}) \quad 1 > 0$ in the current case, such a $\overline{\beta}$ is an element of B if and only if it has at most $m+2 \quad w(\overline{\gamma})$ free variables, and the set of free variables in $\overline{\beta}$ must be $\{1\} \ [\ S(\overline{\gamma}) \ [\ V$ where V is any subset of $[d] \n (\{1\} \ [\ S(\overline{\alpha}))$. Thus in this second case, the contribution of all $\overline{\beta}$

with $\overline{\beta}_1 = \star$ to the coefficient of $h_{\overline{\gamma}}$ in (3.3.24) is

$$\sum_{i=jS(\overline{\gamma})j+1}^{m+2\ \ w(\overline{\gamma})} (\ \ 1)^{m\ \ i}\,\lambda_{i+w(\overline{\gamma})\ \ 1}\begin{pmatrix} d & m & 1 \\ i & jS(\overline{\gamma})j & 1\end{pmatrix} = \sum_{j=jS(\overline{\gamma})j}^{m+1\ \ w(\overline{\gamma})} (\ \ 1)^{m\ \ j\ \ 1}\,\lambda_{j+w(\overline{\gamma})}\begin{pmatrix} d & m & 1 \\ j & jS(\overline{\gamma})j\end{pmatrix}, \tag{3.3.30}$$

where here the index i denotes the total number of free variables in $\overline{\beta}$, the factor of $\binom{d\ \ m\ \ 1}{i\ \ jS(\overline{\gamma})j\ \ 1}$ is the number of ways to choose V (note that $jVj = i \quad jS(\overline{\gamma})j \quad 1$), and in the second expression we made the change of variable $j = i \quad 1$. We conclude that in this case, the coefficient of $h_{\overline{\gamma}}$ in (3.3.24) is given by the sum of (3.3.28) and (3.3.30), which is 0.

Overall, we conclude that the entire RHS of (3.3.24) vanishes for $\alpha \ 2 \ A$, proving the third part of the lemma. □

The next lemma formally constructs N_3 and verifies that it has the required properties, is of acceptable size, and that it takes on nonzero values on a significant part of its domain.

Proof of Lemma 3.3.10. Part (a) follows directly from Lemma 3.3.9(c). Part (b) follows by verifying that for any $t \ 2 \ Z \ n \ f0g$, $\psi(s_1, \ldots, s_d; t) = 0$ for any $s_1, \ldots, s_d \ 2$ $[0, 1]^d$; this means that ψ , which is a combination of partial restrictions of ψ, also vanishes for such t. First suppose that t is a positive integer. Observe that $t \quad 1$ while $s_i \quad \frac{1}{d\ 1}\sum_{j\not=i} s_j \ 2 \ [\ \ 1, 1]$, so each ReLU in the definition of ψ is activated and we get

$$\psi(s_1, \ldots, s_d; t) = \sum_{i=1}^{d}\left[t \quad \left(s_i \quad \frac{1}{d\ \ 1}\sum_{j\not=i} s_j\right)\right] \ dt = \ \sum_{i=1}^{d}\left(s_i \quad \frac{1}{d\ \ 1}\sum_{j\not=i} s_j\right) = 0. \tag{3.3.31}$$

Next suppose that t is a negative integer. Then $t \qquad 1$ while $s_i \quad \frac{1}{d\ 1}\sum_{j\not=i} s_j \ 2 \ [\ \ 1, 1]$, so each ReLU in the definition of h is inactive and we get $\psi(s_1, \ldots, s_d; t) = 0$.

For part (c), observe that by the size bound in Lemma 3.3.9(a) and the fact that ψ contains $O(d)$ ReLUs, the size of N_3 may be bounded by

$$S \quad O(d) \quad (\binom{d}{m}(d \quad m) \quad 3^m + 1) \quad O(d)(\frac{d^{m+1} \quad 3^m}{m!} + 1) \quad d^{m+2} \quad d^{2m}$$

for m larger than some absolute constant.

It remains to prove part (d). For brevity, we will omit the parameter t and just refer to $\psi(0, \ldots, 0, s; t)$ and $\psi\ (0, \ldots, 0, s; t)$ as $\psi(0, \ldots, 0, s)$ and $\psi\ (0, \ldots, 0, s)$. We first compute $\psi(0, \ldots, 0, s)$: for s 2 $[0, 1]$,

$$\psi(0, \ldots, 0, s) = \mathrm{ReLU}(\quad s) + (d \quad 1)\,\mathrm{ReLU}(\frac{1}{d \quad 1} \quad s) = s. \tag{3.3.32}$$

Next, for any $\overline{\beta}$ 2 B, if $w(\overline{\beta}) = j$ for some $0 \quad j \quad m + 1$, then if $\overline{\beta}_d = \star$,

$$\psi_{\overline{\beta}}(0, \ldots, 0, s) \tag{3.3.33}$$

$$= \psi(\underbrace{1, \ldots, 1}_{j}, \underbrace{0, \quad 0}_{d \quad j \quad 1}, s) \tag{3.3.34}$$

$$= j \quad \mathrm{ReLU}\left(\quad 1 + \frac{1}{d \quad 1}(j \quad 1 + s)\right) + (d \quad j \quad 1) \quad \mathrm{ReLU}\left(\frac{1}{d \quad 1} \quad j + \frac{1}{d \quad 1} \quad s\right) \tag{3.3.35}$$

$$+ \mathrm{ReLU}\left(\quad s + \frac{1}{d \quad 1} \quad j\right) \tag{3.3.36}$$

$$= \frac{d \quad j \quad 1}{d \quad 1} \quad (j + s) + \mathrm{ReLU}\left(\quad s + \frac{1}{d \quad 1} \quad j\right) \tag{3.3.37}$$

Note that when s 2 $[0, 1/(d \quad 1)]$, because $j \quad 1$ (as $\overline{\beta}$ 2 B) this simplifies to

$$= \frac{(d \quad j \quad s)j}{d \quad 1}. \tag{3.3.38}$$

On the other hand, if $\overline{\beta}_d$ 2 $f0, 1g$, then

$$\psi_{\overline{\beta}}(0, \ldots, 0, s) = \psi(\underbrace{1, \ldots, 1}_{j}, \underbrace{0, \quad 0}_{d \quad j}) \tag{3.3.39}$$

$$= j \quad \mathrm{ReLU}\left(\quad 1 + \frac{1}{d \quad 1}(j \quad 1)\right) + (d \quad j) \quad \mathrm{ReLU}\left(\frac{1}{d \quad 1} \quad j\right) = \frac{(d \quad j)j}{d \quad 1}. \tag{3.3.40}$$

As there are $\binom{d}{i}\binom{1}{1}$ (resp. $\binom{d}{i}\binom{1}{1}$) partial assignments in $B_{i,j}$ for which $\overline{\beta}_d = \star$ (resp. $\overline{\beta}_d \in \{0,1\}$), we can thus explicitly compute $h(0,\dots,0,s)$ for $s \in [0, 1/(d-1)]$ to be

$$\psi(0,\dots,0,s) \sum_{i=0}^{m} \sum_{j=1}^{m+1} (-1)^{m-i} \binom{d-i-j-1}{m-i-j+1} \left(\binom{d-1}{i-1} \frac{(d-j-s)j}{d-1} + \binom{d-1}{i} \frac{(d-j)j}{d-1} \right). \tag{3.3.41}$$

By Lemma B.2.2, the double sum is equal to sero, so $h(0,\dots,0,s) = h(0,\dots,0,s) = s$ for $s \in [0, 1/(d-1)]$ as claimed. □

3.4 Statistical Query Lower Bound

We prove a superpolynomial SQ lower bound (for general queries as opposed to only correlational or Lipschitz queries) for weakly learning two-hidden-layer ReLU networks under the standard Gaussian.

Theorem 3.4.1. *Fix any $\alpha \in (0,1)$. Any SQ learner capable of learning $\mathrm{poly}(d)$-sized two-hidden-layer ReLU networks under $N(0, \mathrm{Id}_d)$ up to squared loss ϵ (for some sufficiently small $\epsilon = 1/\mathrm{poly}(d)$) using bounded queries of tolerance $\tau \geq 2^{-(\log d)^{\alpha}}$ must use at least $\Omega(2^{(\log d)^{\alpha}}\tau^2) = d^{\omega(1)}\tau^2$ such queries.*

For instance, taking $\alpha = \frac{1}{2}$ gives a slightly subexponential (but super-quasipolynomial) in d query lower bound for queries of tolerance at least inverse quasipolynomial in d.

This theorem is proven using the following key reduction, which adapts the compressed DV lift (Theorem 3.3.3) to the SQ setting.

Theorem 3.4.2. *Let $q = \mathrm{poly}(d)$ be a modulus, and let $m = m(d) = \omega_d(1)$ be a size parameter. Let C be a class of compressible L-hidden-layer $\mathrm{poly}(d)$-sized ReLU networks mapping $\mathbb{Z}_q^d$ to $[0,1]$, and let C^M be the lifted class of $(L+1)$-hidden-layer $d^{(m)}$-sized ReLU networks corresponding to C, mapping $\mathbb{R}^d$ to $\mathbb{R}$ (as in Theorem 3.3.3). Suppose there is an SQ learner A capable of learning C^M over $N(0, \mathrm{Id}_d)$ up to squared loss $d^{-(m)}$ using queries of tolerance τ, where $\tau \geq d^{-(m^2)}$. Then there is an SQ learner B that, using the same number of queries of tolerance $\tau/2$, produces a weak*

predictor $\widetilde{B}$ *for* C *over* $\mathrm{Unif}(\mathsf{Z}_q^d)$ *with advantage* $d^{-(m)}$ *over guessing the constant* $1/2$ *(in expectation over both the data and the internal randomness of* $\widetilde{B}$*).*

Proof. Recall that B is given SQ access to a distribution of pairs (x, y) where x $\mathrm{Unif}(Z_q^d)$ and $y = f(x)$ for an unknown $f \in \mathsf{C}$. A can request estimates $\mathsf{E}[\phi(x, y)] \pm \tau$ for arbitrary bounded queries $\phi : \mathsf{Z}_q^d \times [0, 1] \to [-1, 1]$ and any desired τ. We know that given (x, y), the distribution of $(z, \widetilde{y})$, where $z = z(x)$ is defined by drawing each z_j from $\mathcal{N}(0, 1)$ conditioned on $z_j \in I_{x_j}$ and $\widetilde{y} = \widetilde{y}(y, z)$ is as in Eq. (3.3.12)), is consistent with some $f^{\mathsf{M}} \in \mathsf{C}^{\mathsf{M}}$ except on a region of probability mass at most d^{-9m^2} (recall Eq. (3.3.11)). Suppose we could simulate SQ access to the distribution of $(z, f^{\mathsf{M}}(z))$ using only SQ access to that of $(x, f(x))$. Then by the argument in Theorem 3.3.3, simulating A on the $(z, f^{\mathsf{M}}(z))$ distribution would give us a weak predictor $\widetilde{B}$ for the distribution of $(x, f(x))$, satisfying

$$\mathsf{E}\left[\left(\widetilde{B}(x) - f(x)\right)^2\right] < \mathsf{E}\left[\left(\frac{1}{2} - f(x)\right)^2\right] - d^{-(m)}.$$

What we must describe is how B can simulate A's statistical queries. Say A requests an estimate $\mathsf{E}_z[\phi(z, f^{\mathsf{M}}(z))] \pm \tau$ for some query $\phi : \mathsf{R}^d \times \mathsf{R} \to [-1, 1]$. Consider the query $\widetilde{\phi} : \mathsf{Z}_q^d \times [0, 1] \to [-1, 1]$ given by $\widetilde{\phi}(x, y) = \mathsf{E}_{z(x)}[\phi(z(x), \widetilde{y}(y, z(x)))]$. This function can be computed without any additional SQs, since the distribution of $(z, \widetilde{y}) = (z(x), \widetilde{y}(y, z(x)))$, given (x, y), is fully determined and known to B. Observe that

$$\mathop{\mathsf{E}}_{x,y}\widetilde{\phi}(x, y) = \mathop{\mathsf{E}}_{x,z(x)}[\phi(z(x), \widetilde{y}(y, z(x)))] = \mathop{\mathsf{E}}_{z,\widetilde{y}}[\phi(z, \widetilde{y})]. \tag{3.4.1}$$

We must also account for the difference between $\mathsf{E}_z[\phi(z, f^{\mathsf{M}}(z))]$ and $\mathsf{E}_{z,\widetilde{y}}[\phi(z, \widetilde{y})]$. But because the distributions only differ on a region of mass d^{-9m^2} and ϕ is bounded, we have

$$\left|\mathop{\mathsf{E}}_{z}[\phi(z, f^{\mathsf{M}}(z))] - \mathop{\mathsf{E}}_{z,\widetilde{y}}[\phi(z, \widetilde{y})]\right| - \Theta(d^{-9m^2}) - \frac{\tau}{2} \tag{3.4.2}$$

since we assumed $\tau - d^{-(m^2)}$. Putting together (3.4.1) and (3.4.2), we see that B can simulate A's query ϕ to within tolerance τ by querying $\widetilde{\phi}$ with tolerance $\tau/2$. □

Again, by a padding argument we can obtain a corollary similar to Corollary 3.3.4, for which we omit the formal statement. We will use such an argument in the proof of Theorem 3.4.1.

3.4.1 SQ lower bound via parities

We can obtain an SQ lower bound for two-hidden-layer ReLU networks by lifting the problem of learning parities under U_d, which is well-known to require exponentially many queries. More precisely, we show that an SQ learner for two-hidden-layer ReLU networks would yield an SQ algorithm for the problem of distinguishing an unknown parity from random labels.

Theorem 3.4.3 ([Kea98, BFJ+94]). *Consider an SQ algorithm given SQ access either to the distribution of labeled pairs (x, y) where x U_d and $y = \chi_S(x)$ for an unknown parity χ_S or to the randomly labeled distribution U_d Unif$\{0,1\}$. Any algorithm capable of distinguishing between the two cases with probability $2/3$ using queries of tolerance τ requires at least $\Omega(2^d\tau^2)$ such queries.*

Lemma 3.4.4. *For every S $[d]$, the parity function $\chi_S : \{0,1\}^d$! $\{0,1\}$ can be implemented as a compressible one-hidden-layer ReLU network of $\mathrm{poly}(d)$ size.*

Proof. Recall that $\chi_S(x)$ evaluates to 1 if the Hamming weight of the bits of x in S is odd, and 0 otherwise, so that $\chi_S(x) = \sigma(\sum_{j2S} x_j)$. This satisfies the definition of a compressible one-hidden-layer network with the inner depth-0 network being x 7! $\sum_{j2S} x_j$ and $\sigma(t) = \mathbb{1}[t \text{ is odd}]$. □

We can now supply one proof of Theorem 3.4.1.

First proof of Theorem 3.4.1. Let $m = m(d) = \log^c d$ for $c = \frac{1}{\alpha}$ 1, and let $d' = d^m = 2^{\log^{c+1} d}$, so that $d = 2^{\log^{1/(1+c)} d'}$. By Lemma 3.4.4, the class C of parities on $\{0,1\}^d$ can be implemented by compressible one-hidden-layer $\mathrm{poly}(d)$-sized ReLU networks, and so the lifted class C^M can be implemented by two-hidden-layer $d^{(m)}$-sized ReLU

networks over R^d. A padding argument lets us embed these classes into dimension d'. By using the predictor from Theorem 3.4.2 (with $q = 2$), we could obtain an SQ algorithm capable of distinguishing parities from random labels using queries of tolerance $\tau/2$, assuming $\tau \quad d^{(m^2)} = 2^{\log^{2c+1} d} = 2^{\log^{\frac{2c+1}{c+1}} d'}$. By Theorem 3.4.3, the lower bound for learning parities is $\Omega(2^d\tau^2) = \Omega(2^{2^{\log^{1/(1+c)} d'}}\tau^2)$. Substituting $\alpha = \frac{1}{1+c}$ gives the result. □

But the SQ lower bound obtained this way via parities is somewhat unconvincing since there is a non-SQ algorithm capable of learning the lifted function class obtained from parities. Indeed, suppose we are given examples $(z, f^{\mathsf{M}}(z))$ where f is an unknown parity. We know that whenever z lands in the "good region" G from the proof of Theorem 3.3.3 (which happens with non-negligible probability), we have $f^{\mathsf{M}}(z) = f(\mathrm{sign}(z))N_2(z)$ (recall Eq. (3.3.15)). This means we can simply filter out all $z \notin G$ and form a clean data set of labeled points $(\mathrm{sign}(z), f(\mathrm{sign}(z)))$. The unknown f (and hence f^{M}) can now be learnt by simple Gaussian elimination. In order to give a more convincing lower bound, we now provide an alternative proof based on LWR.

3.4.2 SQ lower bound via the LWR functions

Here we provide an alternative proof of Theorem 3.4.1 using the LWR functions. The hard function class obtained this way is not only *unconditionally* hard for SQ algorithms, it is arguably hard for non-SQ algorithms as well, since LWR is believed to be cryptographically hard.

We begin by stating an SQ lower bound for the LWR functions. This theorem is proven in Section B.3 using a general formulation in terms of pairwise independent function families that may be of independent interest, communicated to us by Bogdanov [Bog21].

Theorem 3.4.5. *Let C_{LWR} denote the $LWR_{n,p,q}$ function class. Any SQ learner capable of learning C_{LWR} up to squared loss $1/16$ under $\mathrm{Unif}(\mathsf{Z}_q^n)$ using queries of tolerance τ requires at least $\Omega(q^{n\ 1}\tau^2)$ such queries.*

The following lemma shows that the LWR functions may be realized as compressible one-hidden-layer ReLU networks.

Lemma 3.4.6. *For every w 2 Z_q^n, the LWR function $f_w : Z_q^n$! Z_p/p can be implemented as a compressible one-hidden-layer ReLU network of size $O(q^2 n)$.*

Proof. By definition, we have $f_w(x) = \frac{1}{p}$b$(w \quad x) \bmod q$e$_p$, which is a compressible one-hidden-layer ReLU network with the inner depth-0 network (i.e., affine function) being w 7! $w \quad x$ and $\sigma(t) = \frac{1}{p}$b$t \bmod q$e$_p$. The size bound follows by observing that for any x 2 Z_q^n, the quantity $w \quad x$ is an integer in f$0, \ldots, q^2 n$g. □

We are ready for an alternative proof of Theorem 3.4.1.

Alternative proof of Theorem 3.4.1. Let n be the security parameter, and fix moduli $p, q \quad 1$ such that $p, q = \mathrm{poly}(n)$ and $p/q = \mathrm{poly}(n)$. Let $d = n$, so that the SQ lower bound from Theorem 3.4.5 is $\Omega(q^n \ ^1) = d \ ^{(d)} = 2^{\tilde{}(d)}$. Let $m = m(d) = \log^c d$ for $c = \frac{1}{\alpha} \quad 1$, and let $d^0 = d^m = 2^{\log^{c+1} d}$, so that $d = 2^{\log^{1/(1+c)} d^0}$. By Lemma 3.4.6, the LWR$_{n,p,q}$ function class C_{LWR} is implementable by one-hidden-layer ReLU networks over Z_q^d of size $\mathrm{poly}(n) = \mathrm{poly}(d)$. The result now follows by Theorem 3.4.2 and the same padding argument as in the proof based on parities. □

3.5 Cryptographic Hardness Based on LWR

In this section we show hardness of learning two-hidden-layer ReLU networks over Gaussian inputs based on LWR. This is a direct application of the compressed DV lift (Theorem 3.3.3) to the LWR problem, which is by definition a hard learning problem over Unif(Z_q^n), or equivalently Unif(Z_q^d) with $d = n$.

Theorem 3.5.1. *Let n be the security parameter, and x moduli $p, q \quad 1$ such that $p, q = \mathrm{poly}(n)$ and $p/q = \mathrm{poly}(n)$. Let $d = n$. Let $c > 0$, $m = m(d) = \log^c d$ and*

$d' = d^m$. *Suppose there exists a* $\mathrm{poly}(d')$*-time algorithm capable of learning* $\mathrm{poly}(d')$*-sized depth-2 ReLU networks under* $N(0, \mathrm{Id}_{d'})$ *up to squared loss* $1/\mathrm{poly}(d')$. *Then there exists a* $\mathrm{poly}(d') = 2^{(\log^{1+c} n)}$ *time algorithm for* $\mathsf{LWR}_{n,p,q}$.

Proof of Theorem 3.5.1. By Lemma 3.4.6, we know that the class $\mathcal{C}_{\mathsf{LWR}}$ is implementable by compressible $\mathrm{poly}(d)$-sized one-hidden-layer ReLU networks over $\mathbb{Z}_q^d$, or, after padding, over $\mathbb{Z}_q^{d'}$. Let $\mathcal{C}^{\mathsf{M}}_{\mathsf{LWR}}$ denote the corresponding lifted class of $\mathrm{poly}(d')$-sized two-hidden-layer ReLU networks, padded to have domain $\mathbb{R}^{d'}$. Applying Corollary 3.3.4 to the assumed learner for $\mathcal{C}^{\mathsf{M}}_{\mathsf{LWR}}$, we obtain a $\mathrm{poly}(d')$-time weak predictor predictor for $\mathcal{C}_{\mathsf{LWR}}$, which readily yields a corresponding distinguisher for the $\mathsf{LWR}_{n,p,q}$ problem. Using the facts that $d' = d^m = 2^{\log^{1+c} d}$ and $d = n$, we may translate $\mathrm{poly}(d')$ into $2^{(\log^{1+c} n)}$, yielding the result. □

Remark 3.5.2. Note that the choice of $m = m(d) = \log^c d$ in Theorem 3.5.1 is purely for simplicity. By picking $m(d) = \omega_d(1)$ to be a suitably slow-glowing function of d, such as $\log d$, we can obtain a running time for the final LWR algorithm that is as mildly superpolynomial as we like.

In addition, as an immediate corollary of Lemma 3.4.6, we also obtain a hardness result for one-hidden-layer networks under $\mathrm{Unif}\{0,1\}^d$, improving on the hardness result of [DV21] (see Theorem 3.4 therein) for *two*-hidden-layer networks under $\mathrm{Unif}\{0,1\}^d$. For this application, we let $d = n \log q = \widetilde{O}(n)$, so that we may identify the domain $\mathbb{Z}_q^n$ with $\{0,1\}^d$ via the binary representation. This also identifies $\mathrm{Unif}(\mathbb{Z}_q^n)$ with $\mathrm{Unif}\{0,1\}^d$.

Corollary 3.5.3. *Let* n, p, q *be such that* $p, q = \mathrm{poly}(n)$ *and* $p/q = \mathrm{poly}(n)$*, and let* $d = n \log q = \widetilde{O}(n)$. *Suppose there exists an e cient algorithm for learning* $\mathrm{poly}(d)$*-sized one-hidden-layer ReLU networks under* U_d *up to squared loss* $1/4$. *Then there exists an e cient algorithm for* $\mathsf{LWR}_{n,p,q}$.

3.6 Hardness of Learning using Label Queries

The main result of this section is to show hardness of learning constant-depth ReLU networks over Gaussians from label queries:

Theorem 3.6.1. *Assume there exists a family of PRFs mapping $f0,1g^d$ to $f0,1g$ implemented by $\mathrm{poly}(d)$-sized L-hidden-layer ReLU networks. Then there does not exist an e cient learner that, given query access to an unknown $\mathrm{poly}(d)$-sized $(L+2)$-hidden-layer ReLU network $f : \mathsf{R}^d \; ! \; \mathsf{R}$, is able to output a hypothesis $h : \mathsf{R}^d \; ! \; \mathsf{R}$ such that $\mathsf{E}_{z \; N(0,\mathrm{Id}_d)}[(h(z) \quad f(z))^2] \quad 1/16$.*

We first recall the classical connection between pseudorandom functions and learning from label queries (also known as membership queries in the Boolean setting), due to Valiant [Val84] (see e.g. [BR17, Proposition 12] for a modern exposition).

Lemma 3.6.2. *Let $C = f f_s g$ be a family of PRFs from $f0,1g^d$ to $f0,1g$ indexed by the key s. Then there cannot exist an e cient learner L that, given query access to an unknown f_s 2 C, satis es*

$$\mathop{\mathsf{P}}_{x,s}[L(x) = f_s(x)] \quad \frac{1}{2} + \frac{1}{\mathrm{poly}(d)},$$

where the probability is taken over the random key s, the internal randomness of A, and a random test point x Unif $f0,1g^d$.

There exist multiple candidate constructions of PRF families in the class TC^0 of constant-depth Boolean circuits built with AND, OR, NOT and threshold (or equivalently majority) gates. Because the majority gate can be simulated by a linear combination of ReLUs similar to N_1 from Lemma 3.3.6, any TC^0_L (meaning depth-L) function $f : f0,1g^d \; ! \; f0,1g$ may be implemented as a $\mathrm{poly}(d)$-sized L-hidden-layer ReLU network (see e.g. [VRPS21, Lemma A.3][3]). Thus we may leverage the following candidate PRF constructions in TC^0 for our hardness result:

[3] Note that what the authors term a depth-$(L + 1)$ network is in fact an L-hidden-layer network in our terminology.

- PRFs in TC^0_4 based on the decisional Diffie-Hellman (DDH) assumption [KL01] (improving on [NR97]), yielding hardness for depth-6 ReLU networks
- PRFs in TC^0 based on Learning with Errors [BPR12, BP14], yielding hardness for depth-$O(1)$ ReLU networks

Note that depth 4 is the shallowest depth for which we have candidate PRF constructions based on widely-believed assumptions, and the question of whether there exist PRFs in TC^0_3 is a longstanding open question in circuit complexity [Raz92, HMP+93, RR97, KL01]. Under less widely-believed assumptions, [BIP+18] have also proposed candidate PRFs in ACC^0_3.

We can now complete the proof of Theorem 3.6.1. Since pseudorandom functions are not necessarily compressibile, we will simply use the original DV lift (Theorem 3.3.5).

Proof of Theorem 3.6.1. Let $f_s : f0,1g^d \; ! \; f0,1g$ be an unknown L-hidden-layer ReLU network obtained from the PRF family by picking the key s at random. Consider the lifted $(L+2)$-hidden-layer ReLU network $f_s^{\mathsf{DV}} : \mathsf{R}^d \; ! \; \mathsf{R}$ from Eq. (3.3.1), given by $f_s^{\mathsf{DV}}(z) = \mathrm{ReLU}(f_s(N_1(z)) \quad N_2^{\emptyset}(z))$, where N_1 and N_2 are from Lemmas 3.3.6 and 3.3.7, and $N_2^{\emptyset}(z) = \sum_j N_2(z_j)$. Suppose there were an efficient learner A capable of learning functions of the form f_s^{DV} using queries. By the DV lift (Theorem 3.3.5), A yields an efficient predictor B achieving small constant error w.r.t. the unknown f_s, contradicting Lemma 3.6.2. We only need to verify that A's query access to f_s^{DV} can be simulated by B. Indeed, suppose A makes a query to f_s^{DV} at a point $z \; 2 \; \mathsf{R}^d$. Then B can make a query to f_s at the point $\mathrm{sign}(z)$ and return $\mathrm{ReLU}(f_s(\mathrm{sign}(z)) \quad N_2^{\emptyset}(z)) = f_s^{\mathsf{DV}}(z)$, as this was the key property satisfied by f_s^{DV}. This completes the reduction and proves the theorem. □

Chapter 4: Statistical-Query Lower Bounds via Functional Gradients

4.1 Introduction

In this chapter we continue a recent line of research exploring the computational complexity of fundamental primitives from the theory of deep learning [GKK19, YS19, DKKZ20, YS20, DGK+20, FCG20]. In particular, we consider the problem of fitting a single nonlinear activation to a joint distribution on $\mathsf{R}^n \quad \mathsf{R}$. When the nonlinear activation is ReLU, this problem is referred to as ReLU regression or agnostically learning a ReLU. When the nonlinear activation is sign and the labels are Boolean, this problem is equivalent to the well-studied challenge of agnostically learning a halfspace [KKMS08].

We consider arguably the simplest possible setting—when the marginal distribution is Gaussian—and give the first statistical-query lower bounds for learning broad classes of nonlinear activations. The statistical-query model is a well-studied framework for analyzing the sample complexity of learning problems and captures most known learning algorithms. For common activations such as ReLU, sigmoid, and sign, we give complementary upper bounds, showing that our results cannot be significantly improved.

Let H be a function class on R^n, and let D be a labeled distribution on $\mathsf{R}^n \quad \mathsf{R}$ such that the marginal on R^n is $D = N(0, I_n)$. We say that a learner learns H under D with error ϵ if it outputs a function f such that

$$\mathop{\mathsf{E}}_{(x,y)\ D}[f(x)y] \quad \max_{h2H} \mathop{\mathsf{E}}_{(x,y)\ D}[h(x)y] \quad \epsilon.$$

One can show that this loss captures 0-1 error in the Boolean case, as well as squared loss in the ReLU case whenever the learner is required to output a nontrivial hypothesis (i.e., a hypothesis with norm bounded below by some constant $c > 0$). (See Sections C.5 and C.6 for details.)

For ReLU regression, we obtain the following exponential lower bound:

Theorem 4.1.1. *Let H_{ReLU} be the class of ReLUs on R^n with unit weight vectors. Suppose that there is an SQ learner capable of learning H_{ReLU} under D with error ϵ using $q(n, \epsilon, \tau)$ queries of tolerance τ. Then for any ϵ, there exists $\tau = n^{(1/\epsilon)^b}$ such that $q(n, \epsilon, \tau) \quad 2^{n^c}\epsilon$ for some $0 < b, c < 1/2$. That is, a learner must either use tolerance smaller than $n^{(1/\epsilon)^b}$ or more than $2^{n^c}\epsilon$ queries.*

Prior work due to Goel et al. [GKK19] gave a quasipolynomial SQ lower bound (with respect to correlational queries) for ReLU regression when the learner is required to output a ReLU as its hypothesis.

For the sigmoid activation we obtain the following lower bound:

Theorem 4.1.2. *Consider the above setup with H_σ, the class of unit-weight sigmoid units on R^n. For any ϵ, there exists $\tau = n^{(\log^2 1/\epsilon)}$ such that $q(n, \epsilon, \tau) \quad 2^{n^c}\epsilon$ for some $0 < c < 1/2$.*

We are not aware of any prior work on the hardness of agnostically learning a sigmoid with respect to Gaussian marginals.

For the case of halfspaces, a result of Kalai et al. [KKMS08] showed that any halfspace can be agnostically learned with respect to Gaussian marginals in time and sample complexity $n^{O(1/\epsilon^4)}$, which was later improved to $n^{O(1/\epsilon^2)}$ [DGJ+10]. The only known hardness result for this problem is due to Klivans and Kothari [KK14a] who gave a quasipolynomial lower bound based on the hardness of learning sparse parity with noise. Here we give the first exponential lower bound:

Theorem 4.1.3. *Consider the above setup with H_{hs}, the class of unit-weight halfspaces on R^n. For any ϵ, there exists $\tau = n^{(1/\epsilon)}$ such that $q(n, \epsilon, \tau) \quad 2^{n^c}\epsilon$ for some xed constant $0 < c < 1/2$.*

Since it takes $\Theta(1/\tau^2)$ samples to simulate a query of tolerance τ, our constraint on τ here can be interpreted as saying that to avoid the exponential query

lower bound, one needs sample complexity at least $\Theta(1/\tau^2) = n^{(1/\epsilon)}$, nearly matching the upper bound of [KKMS08, DGJ+10].

These results are formally stated and proved in Section 4.5. More generally, we show in Section 4.6 that our results give superpolynomial SQ lower bounds for agnostically learning any non-polynomial activation. (See Section C.1 for some discussion of subtleties in interpreting these bounds.)

A notable property of our lower bounds is that they hold for *general* statistical queries. As noted by several authors [APVZ14, VW19a], proving SQ lower bounds for real-valued learning problems often requires further restrictions on the types of queries the learner is allowed to make (e.g., correlational or Lipschitz queries).

Another consequence of our framework is the first SQ lower bound for agnostically learning monomials with respect to Gaussian marginals. In contrast, for the realizable (noiseless) setting, recent work due to Andoni et al. [ADHV19] gave an attribute-efficient SQ algorithm for learning monomials. They left open the problem of making their results noise-tolerant. We show in Section 4.7 that in the agnostic setting, no efficient SQ algorithm exists.

Theorem 4.1.4. *Consider the above setup with H_{mon}, the class of multilinear monomials of degree at most d on $\mathbb{R}^n$. For any $\epsilon \quad \exp(-\Theta(d))$ and $\tau \quad \epsilon^2$, $q(n, \epsilon, \tau)$ $n^{\Omega(d)}\tau^{5/2}$.*

4.1.1 Our Approach

Our approach deviates from the standard template for proving SQ lower bounds and may be of independent interest. In almost all prior work, SQ lower bounds are derived by constructing a sufficiently large family of nearly orthogonal functions with respect to the underlying marginal distribution. Instead, we will use a reduction-based approach:

- We show that an algorithm for agnostically learning a single nonlinear activation

ϕ can be used as a subroutine for learning depth-two neural networks of the form $\psi(\sum_i \phi(w^i \quad x))$ where ψ is any monotone, Lipschitz activation. This reduction involves an application of functional gradient descent via the Frank–Wolfe method with respect to a (nonstandard) convex surrogate loss.

- We apply recent work due to [DKKZ20] and [GGJ+20] that gives SQ lower bounds for learning depth-two neural networks of the above form in the probabilistic concept model. For technical reasons, our lower bound depends on the norms of these depth-two networks, and we explicitly calculate them for ReLU and sigmoid.

- We prove that the above reduction can be performed using only statistical queries. To do so, we make use of some subtle properties of the surrogate loss and the functional gradient method itself.

Our reduction implies the following new relationship between two well-studied models of learning: if concept class C is efficiently agnostically learnable, then the class of monotone, Lipschitz functions of linear combinations of C is learnable in the *probabilistic concept* model due to Kearns and Schapire [KS94b]. We cannot hope to further strengthen the conclusion to *agnostic* learnability of monotone, Lipschitz functions of combinations of C: the concept class of literals *is* agnostically learnable, but we show exponential SQ lower bounds for agnostically learning the class of majorities of literals, i.e., halfspaces (see also [KK14a]).

4.1.2 Related Work

Several recent papers have considered the computational complexity of learning simple neural networks [Bac17, GKKT17a, YS20, FCG20, KK14a, LSSS14, SVWX17, VW19a, GKK19, GGJ+20, DKKZ20]. The above works either consider one-layer neural networks (as opposed to learning single neurons), or make use of discrete distributions (rather than Gaussian marginals), or hold for narrower classes of algorithms

(rather than SQ algorithms). Goel et al. [GKK19] give a quasipolynomial correlational SQ lower bound for proper agnostic learning of ReLUs with respect to Gaussian marginals. They additionally give a similar computational lower bound assuming the hardness of learning sparse parity with noise.

The idea of using functional gradient descent to learn one hidden layer neural networks appears in work due to Bach [Bac17], who considered an "incremental conditional gradient algorithm" that at each iteration implicitly requires an agnostic learner to complete a "Frank–Wolfe step." A key idea in our work is to optimize with respect to a particular convex functional (surrogate loss) in order to obtain SQ learnability for depth-two neural networks *with a nonlinear output activation.* We can then leverage SQ lower bounds for this broader class of neural networks.

Functional gradient descent or gradient boosting methods have been used frequently in learning theory, especially in online learning (see e.g., [Fri01, MBBF00, SF12, BHKL15, Haz16].)

For Boolean functions, the idea to use boosting to learn majorities of a base class appeared in Jackson [Jac97], who boosted a weak parity learning algorithm in order to learn thresholds of parities (TOP). Agnostic, distribution-specific boosting algorithms for Boolean functions have appeared in works due to Kalai and Kanade [KK09] and also Feldman [Fel10]. Agnostic boosting in the context of the SQ model is explored in [Fel12], where an SQ lower bound is given for agnostically learning monotone conjunctions with respect to the uniform distribution on the Boolean hypercube.

The SQ lower bounds we obtain for agnostically learning halfspaces can be derived using one of the above boosting algorithms due to Kalai and Kanade [KK09] or Feldman [Fel10] in place of functional gradient descent, as halfspaces are Boolean functions.

Independent Work Independently and concurrently, Diakonikolas et al. [DKZ20b] have obtained similar results for agnostically learning halfspaces and ReLUs. Rather than using a reduction-based approach, they construct a hard family of Boolean functions. They show that an agnostic learner for halfspaces or ReLUs would yield a learner for this family, which would solve a hard unsupervised distribution-learning problem considered in [DKS17]. Quantitatively, the lower bound they obtain is that agnostic learning of halfspaces or ReLUs up to excess error ϵ using queries of tolerance $n^{-\mathsf{poly}(1/\epsilon)}$ requires at least $n^{\mathsf{poly}(1/\epsilon)}$ queries. These results are technically incomparable with ours. For queries of similar tolerance, our bound of $2^{n^c}\epsilon$ scales exponentially with n whereas theirs only scales polynomially, so that for any constant ϵ our bound is exponentially stronger. But our bound does not scale directly with $1/\epsilon$ (other than via the induced constraint on tolerance, which does scale as $n^{-\mathsf{poly}(1/\epsilon)}$). Our work also extends to general non-polynomial activations, while theirs does not.

4.1.3 Organization

We cover the essential definitions, models and existing lower bounds that we need in the preliminaries. Our main reduction, which says that if we could agnostically learn a single neuron, then we could learn depth-two neural networks composed of such neurons, is set up as follows. In Section 4.3 we explain our usage of functional gradient descent, with Assumption 4.3.1 formally stating the kind of agnostic learning guarantee we require for a single neuron. The main reduction itself is Theorem 4.4.1, the subject of Section 4.4. In Sections 4.5, 4.6 and 4.7 we derive the formal lower bounds which follow as a consequence of our reduction. Finally in Section 4.8, we contrast these lower bounds by also including some simple upper bounds.

4.2 Preliminaries

Notation Let D be a distribution over R^n, which for us will be the standard Gaussian $\mathsf{N}(0, I_n)$ throughout. We will work with the L^2 space $L^2(\mathsf{R}^n, D)$ of functions

from $\mathbb{R}^n$ to $\mathbb{R}$, with the inner product given by $\langle f, g \rangle_D = \mathbb{E}_D[fg]$. The corresponding norm is $\|f\|_D = \sqrt{\mathbb{E}_D[f^2]}$. We refer to the ball of radius R as $B_D(R) = \{f \in L^2(\mathbb{R}^n, D) : \|f\|_D \leq R\}$. We will omit the subscripts when the meaning is clear from context. Given vectors $u, v \in \mathbb{R}^n$, we will refer to their Euclidean dot product by $u \cdot v$ and the Euclidean norm by $\|u\|_2$. Given a function $\ell(a, b)$ we denote its partial derivative with respect to its first parameter, $\frac{\partial \ell}{\partial a}(a, b)$, by $\partial_1 \ell(a, b)$.

A Boolean probabilistic concept, or p-concept, is a function that maps each point x to a random $\{-1, 1\}$-valued label y in such a way that $\mathbb{E}[y|x] = f(x)$ for a fixed function $f : \mathbb{R}^n \to [-1, 1]$, known as its conditional mean function. We will use D_f to refer to the (unique) induced labeled distribution on $\mathbb{R}^n \times \{-1, 1\}$, i.e. we say $(x, y) \sim D_f$ if the marginal distribution of x is D and $\mathbb{E}[y|x] = f(x)$. We also sometimes use $y \sim f(x)$ to say that $y \in \{-1, 1\}$ and $\mathbb{E}[y|x] = f(x)$.

4.2.1 Statistical Query (SQ) Model

A statistical query is specified by a query function $\phi : \mathbb{R}^n \times \mathbb{R} \to [-1, 1]$. Given a labeled distribution D on $\mathbb{R}^n \times \mathbb{R}$, the SQ model allows access to an SQ oracle (known as the STAT oracle in the SQ literature) that accepts a query ϕ of specified tolerance τ, and responds with a value in $[\mathbb{E}_{(x,y) \sim D}[\phi(x, y)] - \tau, \mathbb{E}_{(x,y) \sim D}[\phi(x, y)] + \tau]$. One can interpret the tolerance τ as capturing the notion of sample complexity in traditional PAC algorithms. Specifically, it takes $\Theta(1/\tau^2)$ samples to simulate a query of tolerance τ, and this is sometimes referred to as the estimation complexity of an SQ algorithm.

Let C be a class of Boolean p-concepts over $\mathbb{R}^n$, and let D be a distribution on $\mathbb{R}^n$. We say that a learner learns C with respect to D up to L^2 error ϵ if, given only SQ oracle access to D_f for some unknown $f \in C$, and using arbitrary queries, it is able to output $f : \mathbb{R}^n \to [-1, 1]$ such that $\|f - f\|_D \leq \epsilon$. It is worth emphasizing that a query to D_f takes in a Boolean rather than a real-valued label, i.e. is really of the form $\phi : \mathbb{R}^n \times \{-1, 1\} \to [-1, 1]$. In contrast, a query to a generic distribution

D on R^n R takes in real-valued labels, and in Assumption 4.3.1 we define a form of learning that operates in this more generic setting.

One of the chief features of the SQ model is that one can give strong information theoretic lower bounds on learning a class C in terms of its so-called statistical dimension.

Definition 4.2.1. Let D be a distribution on R^n, and let C be a real-valued or Boolean concept class on R^n. The *average (un-normalized) correlation* of C is defined to be $\rho_D(\mathsf{C}) = \frac{1}{j\mathsf{C}j^2} \sum_{c,c^0 2\mathsf{C}} jhc, c^0 i_D j$. The *statistical dimension on average* at threshold γ, $\mathrm{SDA}_D(\mathsf{C}, \gamma)$, is the largest d such that for all C^0 C with $j\mathsf{C}^0 j$ $j\mathsf{C}j/d$, $\rho_D(\mathsf{C}^0)$ γ.

In the p-concept setting, lower bounds against general queries in terms of SDA were first formally shown in [GGJ+20].

Theorem 4.2.2 ([GGJ+20], Cor. 4.6)**.** *Let D be a distribution on R^n, and let C be a p-concept class on R^n. Say our queries are of tolerance τ, the nal desired L^2 error is ϵ, and that the functions in C satisfy $kf k$ β for all f 2 C. For technical reasons, we will require τ ϵ^2, ϵ $\beta/3$ (see Section C.1 for some discussion). Then learning C up to L^2 error ϵ (we may pick ϵ as large as $\beta/3$) requires at least $\mathrm{SDA}_D(\mathsf{C}, \tau^2)$ queries of tolerance τ.*

A recent result of Diakonikolas et al [DKKZ20] gave the following construction of one-layer neural networks on R^n with k hidden units, i.e. functions of the form $g(x) = \psi(\sum_{i=1}^k a_i \phi(x \quad w_i))$ for activation functions $\psi, \phi : \mathsf{R} \; ! \; \mathsf{R}$ and weights w_i 2 R^n, a_i 2 R.

Theorem 4.2.3 ([DKKZ20])**.** *There exists a class G of one-layer neural networks on R^n with k hidden units such that for some universal constant $0 < c < 1/2$ and $\gamma = n^{(k(c \ 1/2))}$, $\mathrm{SDA}(\mathsf{G}, \gamma)$ 2^{n^c}. This holds for any $\psi : \mathsf{R} \; ! \; [\ 1, 1]$ that is odd, and ϕ 2 $L^2(\mathsf{R}, \mathsf{N}(0, 1))$ that has a nonzero Hermite coe cient of degree greater than $k/2$. Further, the weights satisfy $ja_i j = 1/k$ and $kw_i k_2 = 1$ for all i.*

We will be interested in the following special cases. Full details of the construction and proofs of the norm lower bounds are in Section C.2.

Corollary 4.2.4. *For the following instantiations of G, with accompanying norm lower bound β (i.e. such that $kgk \quad \beta$ for all g 2 G), there exist $\tau = n^{(k)}$ and $\epsilon \quad \tau$ such that learning G up to L^2 error ϵ requires at least 2^{n^c} queries of tolerance τ, for some $0 < c < 1/2$.*

(a) ReLU nets: $\psi = \tanh$, $\phi = \mathrm{ReLU}$. Then $\beta = \Omega(1/k^6)$ (Lemma C.2.4), so we may take $\epsilon = \Theta(1/k^6)$.

(b) Sigmoid nets: $\psi = \tanh$, $\phi = \sigma$. Then $\beta = \exp(\quad O(\sqrt{k}))$ (Lemma C.2.6), so we may take $\epsilon = \exp(\quad \Theta(\sqrt{k}))$.

(c) Majority of halfspaces: $\psi = \phi = \mathrm{sign}$. Being Boolean functions, here $\beta = 1$ exactly, so we may take $\epsilon = \Theta(1)$.

4.2.2 Convex Optimization Basics

Over a general inner product space Z, a function $p : Z \to \mathbb{R}$ is convex if for all α 2 $[0,1]$ and $z, z' \in Z$, $p(\alpha z + (1 \quad \alpha)z') \quad \alpha p(z) + (1 \quad \alpha)p(z')$. We say that s 2 Z is a subgradient of p at z if $p(z+h) \quad p(z) \quad \langle s, h\rangle$. We say that p is β-smoothly convex if for all z, h 2 Z and any subgradient s of p at z,

$$p(z+h) \quad p(z) \quad \langle s, h\rangle \quad \frac{\beta}{2}\|h\|^2 .$$

If there is a unique subgradient of p at z, we simply refer to it as the gradient $\nabla p(z)$. It is easily proven that smoothly convex functions have unique subgradients at all points. Another standard property is the following: for any $z, z' \in Z$,

$$p(z) \quad p(z') \quad \langle \nabla p(z), z \quad z'\rangle \quad \frac{1}{2\beta}\|\nabla p(z) \quad \nabla p(z')\|^2 . \tag{4.2.1}$$

In this chapter we will be concerned with convex optimization using the Frank–Wolfe variant of gradient descent, also known as conditional gradient descent. In

order to eventually apply this framework to improper learning, we will consider a slight generalization of the standard setup. Let $Z^0 \quad Z$ both be compact, convex subsets of our generic inner product space. Say we have a β-smoothly convex function $p : Z \; ! \; \mathsf{R}$, and we want to solve $\min_{z2Z^0} p(z)$, i.e. optimize over the smaller domain, while allowing ourselves the freedom of finding subgradients that lie in the larger Z. The Frank–Wolfe algorithm in this "improper" setting is Algorithm 1.

Algorithm 1 Frank–Wolfe gradient descent over a generic inner product space

Start with an arbitrary $z_0 \; 2 \; Z$.
for $t = 0, \ldots, T$ **do**
  Let $\gamma_t = \frac{2}{t+2}$.
  Find $s \; 2 \; Z$ such that $hs, \quad r\, p(z_t)i \quad \max_{s^0 2 Z^0} hs^0, \quad r\, p(z_t)i \quad \frac{1}{2}\delta\gamma_t C_p$.
  Let $z_{t+1} = (1 \quad \gamma_t)z_t + \gamma_t s$.
end for

The following theorem holds by standard analysis (see e.g. [Jag13]). For convenience, we provide a self-contained proof in Section C.4.

Theorem 4.2.5. *Let $Z^0 \quad Z$ be convex sets, and let $p : Z \; ! \; \mathsf{R}$ be a β-smoothly convex function. Let $C_p = \beta \operatorname{diam}(Z)^2$. For every t, the iterates of Algorithm 1 satisfy*

$$p(z_t) \quad \min_{z^0 2 Z^0} p(z^0) \quad \frac{2C_p}{t+2}(1+\delta).$$

4.3 Functional gradient descent

Let $\ell : \mathsf{R} \quad \mathsf{R} \; ! \; \mathsf{R}$ be a loss function. Given a p-concept f and its corresponding labeled distribution D_f, the population loss of a function $f : \mathsf{R}^n \; ! \; \mathsf{R}$ is given by $L(f) = \mathsf{E}_{(x,y) \; D_f}[\ell(f(x), y)]$. We will view L as a mapping from $L^2(\mathsf{R}^n, D)$ to R, and refer to it as the loss functional. The general idea of functional gradient descent is to try to find an f in a class of functions F that minimizes $L(f)$ by performing gradient descent in function space. When using Frank–Wolfe gradient descent, the key step in every iteration is to find the vector that has the greatest projection along

the negative gradient, which amounts to solving a linear optimization problem over the domain. When F is the convex hull $\mathrm{conv}(H)$ of a simpler class H, this can be done using a sufficiently powerful agnostic learning primitive for H. Thus we can "boost" such a primitive in a black-box manner to minimize $L(f)$.

Let $H \quad L^2(\mathbb{R}^n, D)$ be a base hypothesis class for which we have an agnostic learner with the following guarantee:

Assumption 4.3.1. *There is an SQ learner for H with the following guarantee. Let D be any labeled distribution on $\mathbb{R}^n \quad \mathbb{R}$ such that the marginal on $\mathbb{R}^n$ is $D = N(0, I_n)$. Given only SQ access to D, the learner outputs a function $f \in B(\mathrm{diam}(H)/2)$ such that*

$$\mathop{\mathbb{E}}_{(x,y)\ D}[f(x)y] \quad \max_{h \in H} \mathop{\mathbb{E}}_{(x,y)\ D}[h(x)y] \quad \epsilon$$

using $q(n, \epsilon, \tau)$ queries of tolerance τ.

Notice that we do not *require f to lie in H, i.e. the learner is allowed to be improper, but we do require it to have norm at most $\mathrm{diam}(H)/2$. This is to make the competitive guarantee against H meaningful, since otherwise the correlation can be made to scale arbitrarily with the norm.*

With such an H in place, we define $F = \mathrm{conv}(H)$. We assume that $f \in F$. Our objective will be to agnostically learn F: to solve $\min_{f \in F} L(f)$ in such a way that $L(f) \quad L(f^{\ }) \quad \epsilon$.

To be able to use Frank–Wolfe, we require some assumptions on the loss function ℓ.

Assumption 4.3.2. *The loss function $\ell : \mathbb{R} \quad \mathbb{R} \ ! \ \mathbb{R}$ is β-smoothly convex in its first parameter.*

From this assumption, orresponding properties of the loss functional L now follow. First we establish the subgradient, which will itself be an element of $L^2(\mathbb{R}^n, D)$,

i.e. a function from R^n to R. Let $f, h : \mathsf{R}^n \to \mathsf{R}$. Observe that at for every $x \in \mathsf{R}^n, y \in \mathsf{R}$, the subgradient property of ℓ tells us that

$$\ell(f(x) + h(x), y) \geq \ell(f(x), y) \geq \partial_1 \ell(f(x), y) h(x).$$

Taking expectations over $(x, y) \sim D_f$, this yields

$$\begin{aligned} L(f + h) - L(f) &\geq \mathop{\mathsf{E}}_{(x,y) \sim D_f} [\partial_1 \ell(f(x), y) h(x)] \\ &= \mathop{\mathsf{E}}_{x \sim D} [\mathop{\mathsf{E}}_{y|x} [\partial_1 \ell(f(x), y)] h(x)] \\ &= \langle s, h \rangle, \end{aligned}$$

where

$$s : x \mapsto \mathop{\mathsf{E}}_{y|x} [\partial_1 \ell(f(x), y)] = \mathop{\mathsf{E}}_{y \sim f(x)} [\partial_1 \ell(f(x), y)]$$

is thus a subgradient of L at f. β-smooth convexity is also easily established. Taking expectations over $(x, y) \sim D_f$ of the inequality

$$\ell(f(x) + h(x), y) \leq \ell(f(x), y) \geq \partial_1 \ell(f(x), y) h(x) \geq \frac{\beta}{2} h(x)^2,$$

we get

$$L(f + h) - L(f) \geq \langle s, h \rangle \geq \frac{\beta}{2} \|h\|^2$$

for the same subgradient s. By smooth convexity, this subgradient is unique and so we can say that the gradient of L at f is given by $\nabla L(f) : x \mapsto \mathsf{E}_{y \sim f(x)} [\partial_1 \ell(f(x), y)]$.

Example 4.3.3. The canonical example is the squared loss functional, with $\ell_{\mathsf{sq}}(a, b) = (a - b)^2$, which is 2-smoothly convex. Here the gradient has a very simple form, since $\partial_1 \ell_{\mathsf{sq}}(a, b) = 2(a - b)$, and so

$$\mathop{\mathsf{E}}_{y \sim f(x)} [\partial_1 \ell_{\mathsf{sq}}(f(x), y)] = \mathop{\mathsf{E}}_{y \sim f(x)} [2(f(x) - y)] = 2(f(x) - f(x)),$$

i.e. $\nabla L_{\mathsf{sq}}(f) = 2(f - f)$. In fact, it is easily calculated that

$$\begin{aligned} L_{\mathsf{sq}}(f) = \mathop{\mathsf{E}}_{(x,y) \sim D_f} [(f(x) - y)^2] &= \mathop{\mathsf{E}}_{(x,y) \sim D_f} [f(x)^2] - 2 \mathop{\mathsf{E}}_{(x,y) \sim D_f} [f(x) y] + \mathop{\mathsf{E}}_{(x,y) \sim D_f} [y^2] \\ &= \mathop{\mathsf{E}}_{x \sim D} [f(x)^2] - 2 \mathop{\mathsf{E}}_{x \sim D} [f(x) \mathsf{E}[y|x]] + \mathop{\mathsf{E}}_{(x,y) \sim D_f} [y^2] \\ &= \|f\|^2 - 2\langle f, f \rangle + 1, \end{aligned}$$

It is also useful to note that

$$L_{\mathsf{sq}}(f) \quad L_{\mathsf{sq}}(f\ ) = \mathsf{k}f \quad f\ \mathsf{k}^2 . \tag{4.3.1}$$

4.3.1 Frank–Wolfe using statistical queries

We see that our loss functional is a β-smoothly convex functional on the space $L^2(\mathsf{R}^n, D)$. We can now use Frank–Wolfe if we can solve its main subproblem: finding an approximate solution to $\max_{h2F}\mathsf{h}h, \quad \mathsf{r}\, L(f)\mathsf{i}$, where f is the current hypothesis during some iteration. Since this is a linear optimization objective and $\mathsf{F} = \mathrm{conv}(\mathsf{H})$, this is the same as solving $\max_{h2H}\mathsf{h}h, \quad \mathsf{r}\, L(f)\mathsf{i}$. This is almost the guarantee that Assumption 4.3.1 gives us, but some care is in order. What we have SQ access to is the labeled distribution D_f on $\mathsf{R}^n \quad \mathsf{f} \quad 1\mathsf{g}$. It is not clear that we can rewrite the optimization objective in such a way that

$$\max_{h2H} \mathsf{E}_{x \ D}[\ \ h(x)\, \mathsf{r}\, L(f)(x)] = \max_{h2H} \mathsf{E}_{(x,y^{\emptyset}) \ D}[h(x)y^{\emptyset}] \tag{4.3.2}$$

for some distribution D on $\mathsf{R}^n \quad \mathsf{R}$ *that we can simulate SQ access to*. Naively, we might try to do this by letting D be the distribution of $(x, \quad \mathsf{r}\, L(f)(x))$ for $x \quad D$, so that a query $\phi : \mathsf{R} \quad \mathsf{R} \ ! \ \mathsf{R}$ to D can be answered with $\mathsf{E}_{(x,y^{\emptyset}) \ D}[\phi(x, y^{\emptyset})] = \mathsf{E}_{x \ D}[\phi(x, \quad \mathsf{r}\, L(f)(x))]$. But the issue is that in general $\mathsf{r}\, L(f)(x)$ will depend on $f\ (x)$, which we do not know — all we have access to is D_f .

It turns out that for the loss functions we are interested in, we can indeed find a suitable such D. We turn to the details now.

4.4 Functional gradient descent guarantees on surrogate loss

The functional GD approach applied directly to squared loss would allow us to learn $\mathsf{F} = \mathrm{conv}(\mathsf{H})$ using a learner for H (that satisfied Assumption 4.3.1). But by considering a certain surrogate loss, we can use the same learner to actually learn $\psi \quad \mathsf{F} = \mathsf{f}\psi \quad f \ \mathsf{j} \ f \ \mathsf{2} \ \mathsf{F}\mathsf{g}$ for an outer activation function ψ. This is particularly

useful as we can now capture p-concepts corresponding to functions in F by using a suitable $\psi : \mathsf{R} \; ! \; [\;\; 1, 1]$. For example, the common softmax activation corresponds to taking $\psi = \tanh$.

Assume that $\mathsf{E}[y\mathit{j}x] = \psi(f\;\;(x))$ for some activation $\psi : \mathsf{R} \;\; ! \;\; \mathsf{R}$ which is non-decreasing and λ-Lipschitz. Instead of the squared loss, we will consider the following surrogate loss:

$$\ell_{\mathsf{sur}}(a, b) = \int_0^a (\psi(u) \quad b)du.$$

It is not hard to see that $\ell_{\mathsf{sur}}(a, b)$ is convex in its first parameter due to the non-decreasing property of ψ, and that $\partial_1 \ell_{\mathsf{sur}}(a, b) = \psi(a) \quad b$. In fact it is λ-smoothly convex:

$$\begin{aligned}
&\ell_{\mathsf{sur}}(a + t, b) \quad \ell_{\mathsf{sur}}(a, b) \quad \partial_1 \ell_{\mathsf{sur}}(a, b)t \\
&= \int_0^{a+t} (\psi(u) \quad b)du \quad \int_0^a (\psi(u) \quad b)du \quad (\psi(a) \quad b)t \\
&= \int_a^{a+t} (\psi(u) \quad b)du \quad (\psi(a) \quad b)t \\
&= \int_a^{a+t} (\psi(u) \quad \psi(a))du \\
&\quad \int_a^{a+t} \lambda(u \quad a)du \\
&= \frac{\lambda t^2}{2}.
\end{aligned}$$

The gradient of the surrogate loss functional, $L_{\mathsf{sur}}(f) = \mathsf{E}_{(x,y) \;\; D_\psi \; f}[\ell_{\mathsf{sur}}(f(x), y)]$, is given by

$$\mathit{r}\, L_{\mathsf{sur}}(f) : x \; \mathit{7!} \; \underset{y \;\; \psi(f \;\; (x))}{\mathsf{E}}[\partial_1 \ell_{\mathsf{sur}}(f(x), y)] = \psi(f(x)) \quad \psi(f\;\;(x)),$$

i.e. $\mathit{r}\, L_{\mathsf{sur}}(f) = \psi \quad f \quad \psi \quad f$.

We still need to show that the Frank–Wolfe subproblem can be solved using

access to just $D_{\psi\ f}$. Observe that

$$\begin{aligned}
\mathop{\mathsf{E}}_{x\ D}[\ \ h(x)\, r\, L_{\mathsf{sur}}(f)(x)] &= \mathop{\mathsf{E}}_{x\ D}[h(x)(\psi(f\ \ (x)) \quad \psi(f(x)))] \\
&= \mathop{\mathsf{E}}_{x\ D}\left[h(x)\left(\mathop{\mathsf{E}}_{y\ \ \psi(f\ \ (x))}[y] \quad \psi(f(x))\right)\right] \\
&= \mathop{\mathsf{E}}_{(x,y)\ D_{\psi\ f}}[h(x)(y \quad \psi(f(x)))] \\
&= \mathop{\mathsf{E}}_{(x,y^{\prime})\ \ D}[h(x)y^{\prime}],
\end{aligned}$$

where D is the distribution of $(x, y \quad \psi(f(x)))$ for $(x, y) \quad D_{\psi\ f}$. We can easily simulate SQ access to this using $D_{\psi\ f}$: if ϕ is any query to D, then

$$\mathop{\mathsf{E}}_{(x,y^{\prime})\ \ D}[\phi(x, y^{\prime})] = \mathop{\mathsf{E}}_{(x,y)\ D_{\psi\ f}}[\phi(x, y \quad \psi(f(x)))] = \mathop{\mathsf{E}}_{(x,y)\ D_{\psi\ f}}[\phi^{\prime}(x, y)] \tag{4.4.1}$$

for the modified query $\phi^{\prime}(x, y) = \phi(x, y \quad \psi(f(x)))$. This means we can rewrite the optimization objective to fit the form in Eq. (4.3.2). Thus for our surrogate loss, Assumption 4.3.1 allows us to solve the Frank–Wolfe subproblem, giving us Algorithm 2 for learning F.

Algorithm 2 Frank–Wolfe for solving $\min_{f 2 F} L_{\mathsf{sur}}(f)$

Start with an arbitrary $f_0\ 2\ B(\mathrm{diam}(H)/2)$.
for $t = 0, \ldots, T$ **do**
  Let γ_t be $\frac{2}{t+2}$.
  Let D_t be the distribution of $(x, y \quad \psi(f_t(x)))$ for $(x, y) \quad D_{\psi\ f}$.
  Using Assumption 4.3.1, find $h\ 2\ B(\mathrm{diam}(H)/2)$ such that

$$\mathop{\mathsf{E}}_{(x,y^{\prime})\ \ D_t}[h(x)y^{\prime}] \quad \max_{h^{\prime} 2 H} \mathop{\mathsf{E}}_{(x,y^{\prime})\ \ D_t}[h^{\prime}(x)y^{\prime}] \quad \frac{1}{2}\gamma_t\lambda\,\mathrm{diam}(H)^2$$

  Let $f_{t+1} = (1 \quad \gamma_t)f_t + \gamma_t h$.
end for

Theorem 4.4.1. *Let H be a class for which Assumption 4.3.1 holds, and let $F = \mathrm{conv}(H)$. Given SQ access to $D_{\psi\ f}$ for a known non-decreasing λ-Lipschitz activation ψ and an unknown $f \ \ 2\ F$, suppose we wish to learn $\psi \quad f$ in terms of surrogate*

loss, i.e. to minimize $L_{\mathsf{sur}}(f)$*. Then after* T *iterations of Algorithm 2, we have the following guarantee:*

$$L_{\mathsf{sur}}(f_T) \quad L_{\mathsf{sur}}(f\ ) \quad \frac{4\lambda \operatorname{diam}(H)^2}{T+2}.$$

In particular, we can achieve $L_{\mathsf{sur}}(f_T) \quad L_{\mathsf{sur}}(f\ ) \quad \epsilon$ *after* $T = O(\frac{\lambda \operatorname{diam}(H)^2}{\epsilon})$ *iterations. Assuming our queries are of tolerance* τ*, the total number of queries used is at most* $Tq(n, \epsilon/4, \tau) = O(\frac{\lambda \operatorname{diam}(H)^2}{\epsilon} q(n, \epsilon/4, \tau))$*.*

Proof. By the preceding discussion, the surrogate loss functional is λ-smoothly convex, and Algorithm 2 is a valid special case of Algorithm 1, with $Z = B(\operatorname{diam}(H)/2)$ and $Z^0 = \operatorname{conv}(F)$. Thus the guarantee follows directly from Theorem 4.2.5 (setting $\delta = 1$).

To bound the number of queries, observe that it is sufficient to run for $T = \frac{4\lambda \operatorname{diam}(H)^2}{\epsilon} \quad 2$ rounds. In the t^{th} iteration, we invoke Assumption 4.3.1 with

$$\epsilon^0 = \frac{1}{2}\gamma_t \lambda \operatorname{diam}(H)^2 = \frac{\lambda \operatorname{diam}(H)^2}{t+2} \quad \frac{\lambda \operatorname{diam}(H)^2}{T+2} = \frac{\epsilon}{4}.$$

Since $q(n, \epsilon^0, \tau) \quad q(n, \epsilon/4, \tau)$, the bound follows. □

Lastly, we can show that minimizing surrogate loss also minimizes the squared loss. Observe first that $r\, L_{\mathsf{sur}}(f\ ) = 0$. Thus, applying Eq. (4.2.1) with $z = f \quad$ and $z^0 = f$, we obtain

$$\begin{aligned} L_{\mathsf{sur}}(f) \quad L_{\mathsf{sur}}(f\ ) \quad & \frac{1}{2\lambda} k\, r\, L_{\mathsf{sur}}(f) \quad r\, L_{\mathsf{sur}}(f\ ) k^2 \\ &= \frac{1}{2\lambda} k\psi \quad f \quad \psi \quad f\ k^2 \\ &= \frac{1}{2\lambda}(L_{\mathsf{sq}}(\psi \quad f) \quad L_{\mathsf{sq}}(\psi \quad f\ )), \end{aligned} \tag{4.4.2}$$

where L_{sq} is squared loss w.r.t. $D_{\psi\ f}$ and the last equality is Eq. (4.3.1). In particular, Eq. (4.4.2) implies that $\psi \quad f$ achieves the following L^2 error with respect to $\psi \quad f$:

$$k\psi \quad f \quad \psi \quad f\ k \quad \sqrt{2\lambda\,(L_{\mathsf{sur}}(f) \quad L_{\mathsf{sur}}(f\ ))}. \tag{4.4.3}$$

4.5 Lower bounds on learning ReLUs, sigmoids, and halfspaces

The machinery so far has shown that if we could agnostically learn a single unit (e.g. a ReLU or a sigmoid), we could learn depth-two neural networks composed of such units. Since we have lower bounds on the latter problem, this yields the following lower bounds on the former.

Theorem 4.5.1. *Let $H_{\mathsf{ReLU}} = \{x \mapsto \mathrm{ReLU}(w \cdot x) : \|w\|_2 \le 1\}$ be the class of ReLUs on $\mathbb{R}^n$ with unit weight vectors.*[1] *Suppose that Assumption 4.3.1 holds for H_{ReLU}. Then for any ϵ, there exists $\tau = n^{-(1/\epsilon)^{1/12}}$ such that $q(n, \epsilon, \tau) \ge 2^{n^c}\epsilon$ for some $0 < c < 1/2$.*

Proof. Since all our lower bound proofs are similar, to set a template we lay out all the steps as clearly as possible.

- Consider the class G from Theorem 4.2.3 instantiated with $\psi = \tanh$ (which is 1-Lipschitz, so $\lambda = 1$) and $\phi = \mathrm{ReLU}$. By the conditions on the weights, we see that $G \subseteq \tanh \circ F_{\mathsf{ReLU}}$, where $F_{\mathsf{ReLU}} = \mathrm{conv}(H_{\mathsf{ReLU}})$. This construction has a free parameter k, which we will set based on ϵ.

- By our main reduction (Assumption 4.3.1 and Theorem 4.4.1), we can learn $\tanh \circ F_{\mathsf{ReLU}}$ with respect to L_{sur} up to agnostic error ϵ using $O(\frac{1}{\epsilon}q(n, \frac{\epsilon}{4}, \tau))$ queries of tolerance τ. By Eq. (4.4.3), this implies learning G up to L^2 error $\sqrt{2\epsilon}$.

- We know that learning G should be hard. Specifically, Corollary 4.2.4(a) states that if $\epsilon' = \Theta(1/k^6)$ and the queries are of tolerance $\tau = n^{-\Omega(k)}$, then learning up to L^2 error ϵ' should require 2^{n^c} queries.

[1] We use $\mathrm{ReLU}(w \cdot x)$ for simplicity. Any learner can handle this by doing a bit of flip on its own.

- The loss our reduction achieves is $\epsilon' = \sqrt{2\epsilon}$, so we require $\sqrt{2\epsilon} \geq \Theta(1/k^6)$ for the bound to hold. Accordingly, we pick $k = \Theta(\epsilon^{-1/12})$, so that $\tau = n^{(k)} = n^{(\epsilon^{-1/12})}$.

- Thus we must have $\frac{1}{\epsilon} q(n, \frac{\epsilon}{4}, \tau) \leq 2^{n^c}$. Rearranging and rescaling ϵ gives the result.

□

Theorem 4.5.2. *Let $H_\sigma = \{x \mapsto \sigma(w \cdot x) \mid \|w\|_2 \leq 1\}$, where σ is the standard sigmoid, be the class of sigmoid units on $\mathbb{R}^n$ with unit weight vectors. Suppose that Assumption 4.3.1 holds for H_σ. Then for any ϵ, there exists $\tau = n^{((\log 1/\epsilon)^2)}$ such that $q(n, \epsilon, \tau) \leq 2^{n^c}\epsilon$ for some $0 < c < 1/2$.*

Proof. Very similar to the above. We instantiate G with $\psi = \tanh$, $\phi = \sigma$, and observe that $G \subseteq \tanh \circ \mathrm{conv}(H_\sigma)$ and that $\mathrm{diam}(H_\sigma) \leq 2$. In this case, Corollary 4.2.4(b) tells us that we require $\sqrt{2\epsilon} \geq e^{-(\sqrt{k})}$ for the lower bound to hold, so we pick $k = (\log 1/\epsilon)^2$. The result now follows exactly as before. □

We also obtain a lower bound on the class of halfspaces. The traditional way of phrasing agnostic learning for Boolean functions is in terms of the 0-1 loss, and it is not immediately obvious that the correlation loss guarantee of Assumption 4.3.1 is equivalent. But in Section C.5, we show that with a little care, they are indeed effectively equivalent. Note that for Boolean functions, functional GD is not essential; existing distribution-specific boosting methods [KK09, Fel10] can also give us similar results here.

Theorem 4.5.3. *Let $H_{\mathrm{hs}} = \{x \mapsto \mathrm{sign}(w \cdot x) \mid \|w\|_2 \leq 1\}$ be the class of halfspaces on $\mathbb{R}^n$ with unit weight vectors. Suppose that Assumption 4.3.1 holds for H_{hs}. Then for any ϵ, there exists $\tau = n^{(1/\epsilon)}$ such that $q(n, \epsilon, \tau) \leq 2^{n^c}\epsilon^3$ for some $0 < c < 1/2$.*

Proof. To approximate the sign function using a Lipschitz function, we define $\widetilde{\text{sign}}(x)$ to be -1 for $x \leq -1/k$, 1 for $x \geq 1/k$, and linearly interpolate in between. This function is $(k/2)$-Lipschitz. We claim that $\mathcal{G}$ instantiated with $\psi = \phi = \text{sign}$ satisfies $\mathcal{G} \subseteq \widetilde{\text{sign}} \circ \text{conv}(\mathcal{H}_{\text{hs}})$, with $\text{diam}(\mathcal{G}) = 2$. This is because as noted in Theorem 4.2.3, $\mathcal{G}$ has weights $a_i \geq 0$ and $\|a\|_1 \leq 1/k$, so the sum of halfspaces inside ψ is always a multiple of $1/k$, and $\widetilde{\text{sign}}$ behaves the same as sign.

Theorem 4.4.1 now lets us learn $\mathcal{G}$ up to agnostic error ϵ (and hence L^2 error $\sqrt{2k\epsilon}$, by *Eq.* (4.4.3)) using $O(\frac{k^2}{\epsilon} q(n, \epsilon/4, \tau))$ queries of tolerance τ. By Corollary 4.2.4(c), we only need $\sqrt{2k\epsilon} \leq \Theta(1)$ for the lower bound to hold, so we may take $k = \Theta(1/\epsilon)$ to get a lower bound of 2^{n^c}. Thus $\frac{k^2}{\epsilon} q(n, \epsilon/4, \tau) \geq 2^{n^c}$, and rearrangement gives the result. □

4.6 Lower bounds on learning general non-polynomial activations

Here we extend our lower bounds to general non-polynomial activations $\phi : \mathbb{R} \to \mathbb{R}$, by which we mean functions which have an infinite Hermite series $\phi = \sum_a \widehat{\phi}_a H_a$, where the H_a are the normalized probabilists' Hermite polynomials. We will again work with the class $\mathcal{G}$ from Theorem 4.2.3, instantiated with this ϕ and $\psi = \tanh$. In Section C.2, we define this construction formally, letting g be the inner function and f be $\psi \circ g$.

To apply our framework, we need a norm lower bound on f. In Lemma C.2.1 we show that $\|g\|$ is determined only by k, the number of hidden units (there $k = 2m$), and the Hermite expansion of ϕ. The reason we require an infinite Hermite series for ϕ is so that this lower bound, viewed as a function of k, is nonzero for infinitely many k. This then implies that $f = \tanh \circ g$ must be nonzero for infinitely many k. Its norm can only possibly be a function of ϕ and k. In particular, we may assume that it satisfies a norm lower bound $\|f\| \geq \beta(k)$, where β is a function only of k that is nonzero for infinitely many k. Here we view the dependence on ϕ as constant.

A few remarks are in order as to how such a bound $\beta(k)$ may be quantitatively established. If ϕ is either bounded or exhibits only polynomial growth, then the bound on $\|g\|$ (Lemma C.2.1) gives a corresponding lower bound on $\|f\|$ that is also purely a function of k. If ϕ is bounded, the calculation is straightforward and very similar to the $\phi = \sigma$ case (Lemma C.2.6). If ϕ grows only like a polynomial, then one can use a truncation argument similar to the $\phi = \mathrm{ReLU}$ case (Lemma C.2.4).

By Theorem 4.2.2 and Corollary 4.2.4, our lower bound of 2^{n^c} on learning G holds for $\epsilon \quad \beta(k)/3$. Since we can pick k as we like, let us say that for all sufficiently small ϵ, we can achieve $\epsilon \quad \beta(k)/3$ by taking $k = k(\epsilon) = 3\beta^{-1}(\epsilon)$. The corresponding tolerance is then $\tau = n^{-(k(\epsilon))}$, which is still inverse superpolynomial in n.

We now get the following lower bound on learning $H = \{x \mapsto \phi(w \cdot x) : \|w\|_2 \leq 1\}$, again by the same arguments as in Section 4.5. We assume that $\|\phi\|_{L^2(\gamma)} \leq R$ for some R, so that $\mathrm{diam}(H) \leq 2R$.

Theorem 4.6.1. *Suppose that Assumption 4.3.1 holds for H. Then for all sufficiently small ϵ and $\tau = n^{-(k(\epsilon))}$, $q(n, \epsilon, \tau) \geq 2^{n^c \frac{\epsilon}{R^2}}$ for some $0 < c < 1/2$.*

Proof. We have $G \subseteq \tanh \circ \mathrm{conv}(H)$. By functional GD wrt surrogate loss (Theorem 4.4.1), we see that we can learn G up to L^2 error $\sqrt{2\epsilon}$ using $O(\frac{R^2}{\epsilon} q(n, \epsilon, \tau))$ queries of tolerance τ, but we must have $O(\frac{R^2}{\epsilon} q(n, \epsilon, \tau)) \geq 2^{n^c}$. □

4.7 Lower bounds on learning monomials

In this section we show lower bounds against agnostically learning monomials with respect to the Gaussian, establishing Theorem 4.1.4. Let H_{mon} be the class of all multilinear monomials of total degree d on $\mathbb{R}^n$. Clearly $|H_{\mathrm{mon}}| = \binom{n}{d} = n^{\Theta(d)}$. For any two distinct multilinear monomials f, g, clearly $\langle f, g \rangle = 0$ and moreover $\langle \tanh \circ f, \tanh \circ g \rangle = 0$ as well. Thus the class $G = \tanh \circ H_{\mathrm{mon}}$ consists entirely of orthogonal functions. By [GGJ+20, Lemma 2.6], $\mathrm{SDA}(G, \gamma) \geq |G| \gamma = n^{\Theta(d)} \gamma$.

We still need a norm lower bound on G.

Lemma 4.7.1. *Let $x_S = \prod_{i \in S} x_i$ be an arbitrary degree-d multilinear monomial on $\mathbb{R}^n$, where $S \subseteq [n]$ is a subset of size d. Then $\|\tanh x_S\|_1 \geq \exp(-\Theta(d))$.*

Proof. Observe first that $\|x_S\|_1 = 1$. By Paley–Zygmund, we have

$$\mathbb{P}[x_S^2 \geq \theta\, \mathbb{E}[x_S^2]] \geq (1-\theta)^2 \frac{\mathbb{E}[x_S^2]^2}{\mathbb{E}[x_S^4]}.$$

By picking $\theta = 1/2$, say, and using the fact that by Gaussian hypercontractivity,

$$\frac{\mathbb{E}[x_S^2]^2}{\mathbb{E}[x_S^4]} = \prod_{i \in S} \frac{\mathbb{E}[x_i^2]^2}{\mathbb{E}[x_i^4]} \geq \exp(-\Theta(d)),$$

we get that $\mathbb{P}[|x_S| \geq 1/2] \geq \exp(-\Theta(d))$.

Now since tanh is monotonic and odd, we have

$$\mathbb{E}[\tanh(x_S)^2] \geq \tanh(1/2)^2\, \mathbb{P}[|x_S| \geq 1/2] \geq \exp(-\Theta(d)).$$

□

By Theorem 4.2.2 with $\beta = \exp(-\Theta(d))$, we get that for any $\epsilon \geq \exp(-\Theta(d))$ and using queries of tolerance $\tau \leq \epsilon^2$, learning G up to L^2 error ϵ takes at least $\mathrm{SDA}(G, \tau^2) \geq n^{\Omega(d)}\tau^2$ queries.

Now we can use the same arguments as in Section 4.5 to prove the following.

Theorem 4.7.2. *Suppose that Assumption 4.3.1 holds for H_{mon}. Then for any $\epsilon \geq \exp(-\Theta(d))$ and $\tau \leq \epsilon^2$, $q(n, \epsilon, \tau) \geq n^{\Omega(d)}\tau^{5/2}$.*

Proof. Observe that $G \in \tanh \circ \mathrm{conv}(H_{\mathrm{mon}})$, and $\mathrm{diam}(H_{\mathrm{mon}}) \leq 2$. Using the surrogate loss with $\psi = \tanh$, Assumption 4.3.1 and Theorem 4.4.1 tell us that we can learn $\tanh \circ \mathrm{conv}(H_{\mathrm{mon}})$ up to L^2 error $\sqrt{2\epsilon}$ (again by Eq. (4.3.1)) in $O(\frac{1}{\epsilon} q(n, \epsilon, \tau))$ queries of tolerance τ. By our lower bound for G, we must have $\frac{1}{\epsilon} q(n, \epsilon, \tau) \geq n^{\Omega(d)}\tau^2$, or $q(n, \epsilon, \tau) \geq n^{\Omega(d)}\tau^{5/2}$ (since $\epsilon \geq \sqrt{\tau}$). □

4.8 Upper bounds on learning ReLUs and sigmoids

We use a variant of the classic low-degree algorithm ([LMN93]; see also [KKMS08]) to provide simple upper bounds for agnostically learning ReLUs and sigmoids. With respect to $D = N(0, I_n)$, the δ-approximate degree of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the smallest d such that there exists a degree-d polynomial p satisfying $\|f - p\|_1 \le \delta$. We show that for any class of δ-approximate degree d, picking $\delta = O(\epsilon)$ and simply estimating the Hermite coefficients of $x \mapsto \mathbb{E}[y|x]$ up to degree d yields an agnostic learner up to error ϵ, one that satisfies Assumption 4.3.1. We assume bounded labels, say $y \in [-C, C]$ for some constant C.

Let D be a distribution on $\mathbb{R}^n \times \mathbb{R}$ such that the marginal on $\mathbb{R}^n$ is $N(0, I_n)$. Let $f_{\mathsf{cmf}}(x) = \mathbb{E}[y|x]$ denote the conditional mean function of D, and note that $\|f_{\mathsf{cmf}}\|_2 \le C$. Observe that for any f, the correlation $\mathbb{E}_{(x,y) \sim D}[f(x)y]$ equals $\langle f, f_{\mathsf{cmf}} \rangle$. Let H be a hypothesis class with δ-approximate degree d (δ to be determined), and let $R = \mathrm{diam}(H)/2$. Let $h_{\mathsf{opt}} \in H$ achieve $\max_{h \in H} \langle h, f_{\mathsf{cmf}} \rangle$.

Our algorithm will be based on approximating the low-degree Hermite coefficients of f_{cmf}, which is equivalent to performing polynomial L^2 regression. It is well-known that in this context, where d is the δ-approximate degree, polynomial L^1 regression up to degree d gives a squared loss guarantee of δ [KKMS08]. But we will not be able to use this result directly since what we seek is a correlation guarantee. Instead, our approach will involve a sequence of inequalities relating the correlation achieved by f_{cmf}, h_{opt}, and their degree-d approximations. A slight subtlety to keep in mind is that correlation can always be increased by scaling the function. This means that wherever scaling is possible, we have to take some care to rescale functions to have the maximum allowed norm, R.

Let $h_{\mathsf{opt}}^{=d}$ and $f_{\mathsf{cmf}}^{=d}$ be the Hermite components of degree at most d of h_{opt} and f_{cmf} respectively. Let $\tilde{f}_{\mathsf{cmf}}^{=d} = \frac{R}{\|f_{\mathsf{cmf}}^{=d}\|_2} f_{\mathsf{cmf}}^{=d}$. Among polynomials of degree d in $B(R)$, it is easy to see that $\tilde{f}_{\mathsf{cmf}}^{=d}$ maximizes $\langle f, f_{\mathsf{cmf}} \rangle$, so that

$$\langle \tilde{f}_{\mathsf{cmf}}^{=d}, f_{\mathsf{cmf}} \rangle \ge \langle h_{\mathsf{opt}}^{=d}, f_{\mathsf{cmf}} \rangle.$$

Our agnostic learner will look to approximate $\tilde{f}_{\mathsf{cmf}}^{\,d}$ by outputting p defined as follows. Suppose $f_{\mathsf{cmf}} = \sum_{I \in \mathbb{N}^n} \alpha_I H_I$, where H_I is the multivariate Hermite polynomial of index I. For each I of total degree at most d, which we denote as $|I| \leq d$, let β_I be our estimate of $\alpha_I = \langle f_{\mathsf{cmf}}, H_I \rangle$ to within tolerance τ (to be determined). This can be done using $n^{O(d)}$ queries of tolerance τ. Let $\tilde{f} = \sum_{|I| \leq d} \beta_I H_I$, and finally let $p = \frac{R}{\|\tilde{f}\|}\tilde{f}$. We have

$$\begin{aligned}
\left\| \frac{\tilde{f}_{\mathsf{cmf}}^{\,d}}{\left\|\tilde{f}_{\mathsf{cmf}}^{\,d}\right\|} - p \right\|^2 &= R^2 \left\| \frac{f_{\mathsf{cmf}}^{\,d}}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|} - \frac{\tilde{f}}{\left\|\tilde{f}\right\|} \right\|^2 \\
&= R^2 \left\| \frac{f_{\mathsf{cmf}}^{\,d} - \tilde{f}}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|} + \tilde{f}\left(\frac{1}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|} - \frac{1}{\left\|\tilde{f}\right\|} \right) \right\|^2 \\
&\quad 2R^2 \left(\frac{\left\|f_{\mathsf{cmf}}^{\,d} - \tilde{f}\right\|^2}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|^2} + \left\|\tilde{f}\right\|^2 \left(\frac{1}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|} - \frac{1}{\left\|\tilde{f}\right\|} \right)^2 \right) \\
&= 2R^2 \left(\frac{\left\|f_{\mathsf{cmf}}^{\,d} - \tilde{f}\right\|^2}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|^2} + \left(\frac{\left\|f_{\mathsf{cmf}}^{\,d}\right\| - \left\|\tilde{f}\right\|}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|} \right)^2 \right) \\
&\quad 4R^2 \frac{\left\|f_{\mathsf{cmf}}^{\,d} - \tilde{f}\right\|^2}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|^2} \qquad \text{(triangle ineq.)} \\
&\quad \frac{4R^2 n^d \tau^2}{\left\|f_{\mathsf{cmf}}^{\,d}\right\|^2}, \qquad (4.8.1)
\end{aligned}$$

since $\|\tilde{f} - f_{\mathsf{cmf}}^{\,d}\| \leq n^{d/2}\tau$.

We claim that we can assume WLOG that $\left\|\tilde{f}_{\mathsf{cmf}}^{\,d}\right\| \geq \epsilon/(2R)$. Indeed, we know $\max_{h \in H} |\langle h, f_{\mathsf{cmf}} \rangle| = |\langle h_{\mathsf{opt}}, f_{\mathsf{cmf}} \rangle|$ and also $\|h_{\mathsf{opt}} - h_{\mathsf{opt}}^{\,d}\| \leq \delta$. This implies that

$$R\|\tilde{f}_{\mathsf{cmf}}^{\,d}\| = |\langle \tilde{f}_{\mathsf{cmf}}^{\,d}, f_{\mathsf{cmf}} \rangle| \geq |\langle h_{\mathsf{opt}}^{\,d}, f_{\mathsf{cmf}} \rangle| \geq |\langle h_{\mathsf{opt}}, f_{\mathsf{cmf}} \rangle| - C\delta,$$

where the last inequality is Cauchy–Schwarz. If $|\langle h_{\mathsf{opt}}, f_{\mathsf{cmf}} \rangle| \leq \epsilon$ then 0 is a valid agnostic learner. Therefore, we can assume that $|\langle h_{\mathsf{opt}}, f_{\mathsf{cmf}} \rangle| \geq \epsilon$. Choosing $\delta = \frac{\epsilon}{2C}$, this means $\left\|\tilde{f}_{\mathsf{cmf}}^{\,d}\right\| \geq \epsilon/(2R)$.

By Eq. (4.8.1), we then have

$$k\tilde{f}^{\,d}_{\mathsf{cmf}} \quad pk \quad \frac{4Rn^{d/2}\tau}{\epsilon}. \tag{4.8.2}$$

Now observe that

$$\begin{aligned} hp, f_{\mathsf{cmf}}i &= h\tilde{f}^{\,d}_{\mathsf{cmf}}, f_{\mathsf{cmf}}i + hp \quad \tilde{f}^{\,d}_{\mathsf{cmf}}, f_{\mathsf{cmf}}i \\ &\quad hh^{d}_{\mathsf{opt}}, f_{\mathsf{cmf}}i \quad \frac{4RCn^{d/2}\tau}{\epsilon} && \text{(Eq. (4.8.2) and Cauchy–Schwarz)} \\ &= hh_{\mathsf{opt}}, f_{\mathsf{cmf}}i + hh^{d}_{\mathsf{opt}} \quad h_{\mathsf{opt}}, f_{\mathsf{cmf}}i \quad \frac{4RCn^{d/2}\tau}{\epsilon} \\ &\quad hh_{\mathsf{opt}}, f_{\mathsf{cmf}}i \quad \frac{\epsilon}{2} \quad \frac{4RCn^{d/2}\tau}{\epsilon}. && \text{(Cauchy–Schwarz, and using } \delta C = \epsilon/2) \end{aligned}$$

Setting $\tau = \frac{\epsilon^2}{8RCn^{d/2}}$ gives us the desired result, namely that $hp, f_{\mathsf{cmf}}i \quad hh_{\mathsf{opt}}, f_{\mathsf{cmf}}i \quad \epsilon$. Thus we have the following theorem.

Theorem 4.8.1. *The class H_{ReLU} can be agnostically learned up to correlation ϵ (in the sense of Assumption 4.3.1) using $n^{O(\epsilon^{-4/3})}$ queries of tolerance $n^{-(\epsilon^{-4/3})}\epsilon$. Similarly, H_σ can be learned using $n^{\tilde{O}(\log^2 1/\epsilon)}$ queries of tolerance $n^{-\tilde{O}(\log^2 1/\epsilon)}\epsilon^2$.*

Proof. Approximating the Hermite coefficients of degree at most d takes $n^{O(d)}$ queries of tolerance $n^{-(d)}\epsilon$. As we show in Section C.3, the δ-approximate degree of unit-weight ReLUs is $O((1/\delta)^{4/3})$ and for unit-weight sigmoids it is $\tilde{O}(\log^2 1/\delta)$. The guarantees follow by the argument in the preceding discussion. □

We note that our lower bounds for ReLUs and sigmoids were for queries of tolerance $n^{-(\epsilon^{-1/12})}$ and $n^{-(\log^2 1/\epsilon)}$ respectively, which nearly matches these upper bounds.

Chapter 5: The Polynomial Method is Universal for Distribution-Free Correlational SQ Learning

5.1 Introduction

A successful and general approach to distribution-free learning in both Valiant's PAC model [Val84] and Kearns *et al.*'s agnostic model [KSS94] has been the so-called "polynomial method." In this approach, a Boolean function is approximated by a low-degree polynomial with respect to various losses such as the 0/1 loss of the sign of the polynomial or a uniform approximation by the polynomial itself. This representation in turn leads to algorithms whose complexity is typically exponential in the degree of the approximating polynomial.

Several researchers have pointed out that the polynomial method captures the best-known distribution-free learners for Boolean function classes with the exception of classes that include parities [HS07]. It is used for example in the algorithm of Klivans and Servedio [KS04] for PAC-learning DNF formulas as well as the L^1 polynomial regression algorithm of Kalai *et al.* [KKMS08] for agnostically learning conjunctions.

In this work we give a complete characterization of (correlational) statistical query learning in the distribution-free PAC or agnostic setting in terms of the threshold degree or approximate degree of the unknown function class. This shows that unless we can find new algorithms that lie outside the SQ model, our only approach for distribution-free learning is to construct low-degree approximating polynomials.

To be more concrete, let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function. The *threshold degree* $\deg(f)$ of f is the least degree of a polynomial p such that $f(x) = \text{sign}(p(x))$ for all x. The pointwise ϵ-*approximate degree* $\widetilde{\deg}_\epsilon(f)$ of f is the least degree of a polynomial p such that $|f(x) - p(x)| \leq \epsilon$ for all x. It is well-known that for any class C of functions with threshold degree bounded by d, C can

be viewed as a class of halfspaces over an $n^{O(d)}$-dimensional space (corresponding to the monomials up to degree d), and classic halfspace learners then yield distribution-free PAC learners for C that run in $n^{O(d)}$ time. Similarly in the agnostic setting, the results of [KKMS08] imply (as noted for example in [KS10, Prop 2.1]) that for a class C with ϵ-approximate degree bounded by d, degree-d L^1 polynomial regression yields a distribution-free agnostic learner (with error $\mathsf{opt} + \epsilon$) running in time $n^{O(d)}$. This is since pointwise or L^1 approximation implies L^2 approximation (as required by [KKMS08]) with respect to all distributions.

Our main results can now be stated as follows. Formal definitions may be found in the preliminaries.

Theorem 5.1.1. *Let C be a Boolean function class satisfying some mild conditions (namely being closed under pattern restrictions, see Definition 5.2.12), with threshold degree $\Omega(d)$. Any distribution-free PAC learner for C using only correlational statistical queries of tolerance $\tau < \frac{1}{10}$ requires at least $2^{\Omega(d)}\tau^2$ queries in order to learn C up to error $\frac{1}{3}$.*

Theorem 5.1.2. *Let C be a Boolean function class as above (closed under pattern restrictions), with $\frac{1}{2}$-approximate degree $\Omega(d)$. Any distribution-free agnostic learner for C using only correlational statistical queries of tolerance $\tau < \frac{1}{10}$ requires at least $2^{\Omega(d)}\tau^2$ queries in order to agnostically learn C up to excess error $\frac{1}{100}$, i.e. true error $\mathsf{opt} + \frac{1}{100}$.*

Both these lower bounds match (up to logarithmic factors in the exponent) the upper bounds given by the polynomial method that were mentioned earlier.

As one example of a corollary, we obtain essentially tight lower bounds for the important problem of agnostically learning conjunctions (likewise disjunctions). Specifically, it is a classic result [NS94, Pat92] that the class of conjunctions on $\{-1,1\}^n$ has ϵ-approximate degree $\Theta(\sqrt{n})$ for all constant $\epsilon < 1$, and this yields a $2^{\Omega(\sqrt{n})}$ correlational SQ lower bound by the above theorem.

We point out that a slightly different characterization of CSQ PAC learnability in terms of the existence of low-weight threshold representations had been given in [Fel08] (see Theorem 5.4 therein). Combining this with work by Sherstov [She08b, She11] on the relationships between various complexity measures of Boolean function classes as well as the links between learning and communication complexity, one could already obtain a qualitatively equivalent version of Theorem 5.1.1. Similarly but less directly, the ideas used to obtain Theorem 5.1.2 are arguably implicit in [Fel12] in combination with [She08b, She11]. Thus the primary appeal of our results lies less in their novelty and more in the following factors: (a) the statements are clean and explicitly reveal the dependence on both the degree d and the tolerance τ; (b) the proofs are simple, largely self-contained (without reference to communication complexity results), and explicitly state the hard instances for which the lower bound is obtained.

Both our theorems are based on a more general theorem that constructs a hard family of functions from a single function f and a distribution μ which "orthogonalizes" it up to degree d, i.e. under which all degree-d or lower moments of f vanish. This condition has been used in prior works for proving lower bounds for learning neural networks with respect to Gaussian distributions (e.g. [DKKZ20, DKZ20a]).

This general theorem crystallizes a basic principle of SQ lower bounds, saying that in the SQ setting, the hard distributions for a class of functions are the ones that orthogonalize them, i.e. zero out their low-degree moments:

Theorem 5.1.3. *Let $f : f \ 1, 1g^{n/2} \ ! \ R$ be a a Boolean function, and let μ be a distribution that orthogonalizes f up to degree $d \ 1$. There exists a family A of function-distribution pairs on $f \ 1, 1g^n$ where each function in the family is of the form $x \ 7! \ f(x_V \ w)$ for some $V \ [n]$ of size $n/2$ and $w \ 2 \ f \ 1, 1g^{n/2}$, such that any distribution-free PAC learner for A using only correlational statistical queries of tolerance $\tau < \frac{1}{10}$ requires at least $\Omega(2^d\tau^2)$ queries in order to learn C up to error $\frac{1}{3}$. (Here x_V denotes the vector of x_i for $i \ 2 \ V$, and denotes the bitwise XOR.)*

The proof is a simple application of the pattern matrix method due to Sherstov [She11]. As a partial converse, it is not hard to see that distributions under which the target functions have significant low-degree moments allow Fourier-theoretic methods (e.g. the "low degree algorithm") to succeed. We view this result as providing an additional, more general sense in which the polynomial method is a "universal" approach to learning.

One unusual feature of our results is that rather than constructing a single hard distribution under which a large class of functions is hard to learn, our hard families involve a set of function-distribution pairs, where each function is effectively accompanied by its own hard distribution. A distribution-free learner should, of course, succeed equally well on such a family.

Our results also have consequences for the hardness of approximation by linear classes of classes with high threshold degree.

Theorem 5.1.4. *Let C be a Boolean function class closed under pattern restrictions, with threshold degree $\Omega(d)$. Consider the linear hypothesis class with an arbitrary N-dimensional embedding ψ and a norm bound B:*

$$H_{\psi,B} = \{x \mapsto \langle \psi(x), w \rangle : \|w\|_2 \leq B\}.$$

There exists a family A of function-distribution pairs, with the functions lying in C, such that

$$\max_{(f,D)\in A} \min_{h\in H_{\psi,B}} \mathbb{E}_{x\sim D}\left[(f(x) - h(x))^2\right] > \frac{1}{2} - B^{-\Omega(d)}\sqrt{N}2^{-\Omega(d)}.$$

As one application, we see that for all classes with threshold degree $d = \omega(\log n)$ (which includes most classes, e.g. conjunctions, intersections of halfspaces, and AC^0), so that 2^d is superpolynomial, even weak approximation is impossible unless $B^{-\Omega(d)}\sqrt{N} = 2^{-\Omega(d)}$. In particular, weak approximation is impossible with $B, N = \mathrm{poly}(n)$. This generalizes earlier results proved for classes such as conjunctions via the approximate rank [KS10].

Our approach builds closely on the recent work of [MS22], in which the authors introduce a new measure called the "variance" of a class of labeled distributions, and use it to show CSQ lower bounds as well as a host of hardness of approximation results for the class. In this chapter we use the term "correlational variance" to refer to this quantity for clarity. They use this new framework to show an essentially-tight $2^{(n^{1/3})}$ CSQ lower bound for learning DNF formulas by applying technical results of Razborov and Sherstov [RS10]. Our main contribution is to simplify their approach and extend this framework to the more general setting of orthogonalizing distributions. We can then leverage known lower bounds on the threshold and approximate degree of function classes. For example, we can obtain the $2^{(n^{1/3})}$ lower bound of [MS22] by using our main theorem with the classical $\Omega(n^{1/3})$ lower bound due to Minsky and Papert from the 60s [MP69]. This deviates from established frameworks for proving CSQ lower bounds, which generally use variants of the SQ dimension (see [Fel16, Rey20] for surveys).

5.1.1 Related and prior work

There has been a wealth of work on lower bounds for learning various Boolean function classes in various settings, and we can only hope to survey a small slice of it. All Boolean functions in what follow will be assumed to be on n bits and poly(n)-sized.

Cryptographic lower bounds One of the original lines of work on the hardness of learning was based on cryptographic assumptions (see [KV94b, Chap. 6] for a textbook reference). Valiant in his original paper [Val84] observed that pseudorandom families, and the function classes that can compute them, are inherently hard to learn in polynomial time. This was developed further by [KV94a], who proved hardness of polynomial-time learning for Boolean formulas and finite automata based on assumptions underlying public-key cryptography. Another important early work was [Kha93], which showed a quasipolynomial time lower bound for learning AC^0

even under the uniform distribution (matching [LMN93]) based on secure pseudo-random generators. For intersections of halfspaces, [KS09] showed the hardness of polynomial-time learning assuming the security of lattice-based cryptosystems.

Lower bounds on halfspace-based methods Historically one of the chief approaches to learning various classes has been the use of halfspace-based learners such as Perceptron, SVM, and linear programming, including kernel methods and in particular the polynomial method. Measures such as approximate rank [BdW01], dimension and margin complexity [Vap00, CST+00, BDES02, LMSS07] and their very recent probabilistic variants [KMS20] are known to characterize inherent limitations of this approach. In [KS10], an approximate rank bound of $2^{\Omega(\binom{p}{n})}$ was shown for disjunctions, and in [RS10] a dimension complexity bound of $2^{\Omega(n^{1/3})}$ was shown for DNFs. Recent nearly-optimal lower bounds on threshold and approximate degree [She13, BT19, SW19] for classes such as intersections of halfspaces and AC^0, as well as degree-weight tradeoffs [STT12], may also be considered to fall into this line of work.

Complexity-theoretic lower bounds In more recent times there has been a line of work [DLSS14, Dan16b, DS16] establishing hardness of distribution-free learning using complexity theoretic assumptions such as the hardness of refuting random k-SAT. These works establish hardness of polynomial-time learning for DNF formulas and intersections of $\omega(\log n)$ halfspaces, and hardness of agnostic learning for conjunctions, halfspaces, and parities.

SQ lower bounds The SQ model is the most general one in which unconditional lower bounds have been shown for many classes. The canonical result here is the SQ lower bound of $2^{\Omega(n)}$ for learning parities under the uniform distribution [Kea98, BFJ+94], which introduced the SQ dimension. This measure has since been considerably generalized [Fel12, FGR+17, Fel17]. As mentioned in the introduction,

other important works in this overall line are those of [Fel08, Fel12], which provided characterizations and lower bounds for distribution-free CSQ learnability in both the PAC and agnostic settings. These can in turn be related to other commonly studied complexity measures using the far-reaching results of [She08b, She11]. Our approach builds on the slightly different work of [MS22], which proved SQ lower bounds in terms of a measure called variance, variants of which had been considered before in [SSSS17, Sha18b].

In the distribution-specific agnostic setting, the polynomial method takes the form of L^1 polynomial regression, introduced by [KKMS08] as a universal approach to agnostic learning. The results of [DSFT+14, DKPZ21] later showed matching SQ lower bounds in terms of the L^1 approximate degree of a concept class, establishing this quantity as an appropriate measure of agnostic learnability (at least over the uniform distribution on the hypercube or the standard Gaussian). Our results may be seen as analogs of these results for distribution-free PAC and agnostic learnability.

SQ vs CSQ The relationship between the general SQ model and the CSQ restriction has been a topic of much interest. In the distribution-specific setting, the two are unconditionally equivalent [BF02]. More generally, the two are known to be equivalent when either the distribution is known, or sampling or nonuniformity is allowed [Fel08]. But somewhat surprisingly, the two are not in general equivalent in the distribution-free setting: [Fel08] showed that halfspaces are not efficiently CSQ-learnable even though they are always SQ-learnable. Thus our CSQ lower bounds do not readily extend to the general SQ model, and a very compelling direction for future work is to investigate whether such an extension is possible.

The polynomial method and known upper bounds Essentially the only known distribution-free approach to learning most rich Boolean classes remains the polynomial method, i.e. representing Boolean functions as polynomial threshold functions (see [HS07] for a survey) or, for agnostic learning, pointwise approximation (which in

particular implies L^1 approximation [KKMS08] with respect to any distribution). A notable exception in the perfectly realizable case is the use of (linear) algebraic methods for learning classes such as parities or more generally polynomials over finite fields [HS07]. It is significant that the former category of methods may all be implemented using simple linear programming and made to work in the SQ setting, whereas the latter category is the prime (and essentially only) example of PAC algorithms that are not SQ.

Improving upon the polynomial method has long been an elusive goal for many important classes. Indeed, in [ST17] the authors propose a change of perspective where rather than seek an efficient running time, the goal of a distribution-free learner is merely to run with nontrivial savings over exponential time, namely in time $2^{n - s(n)}$ where $s(n) = \omega(\log n)$. For AC^0 circuits of size M and depth d, they give such an algorithm with running time $2^{n - (n/(\log M)^{d-1})}$.

5.2 Preliminaries

Notation Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function and D be a distribution on $\{-1,1\}^n$. We will sometimes view such functions as members of the L^2 space $L^2(\{-1,1\}^n, D)$, with the inner product given by $\langle f, g \rangle_D = \mathsf{E}_D[fg]$. When we use just $\langle f, g \rangle$ without any subscript, we will mean the inner product with respect to the uniform distribution on $\{-1,1\}^n$. We denote by $f(D)$ the labeled distribution (on $\{-1,1\}^n \times \{-1,1\}$) of $(x, f(x))$ for $x \sim D$. We will generally denote unlabeled distributions on $\{-1,1\}^n$ by non-calligraphic letters such as D or the Greek μ, and labeled distributions on $\{-1,1\}^n \times \{-1,1\}$ by calligraphic $\mathcal{D}$ (unless it can be described as $f(D)$ for some f, D).

We use $[n]$ to denote $\{1, \ldots, n\}$. Given $x \in \{-1,1\}^n$ and $V \subseteq [n]$, we denote by x_V the vector of x_i for $i \in V$. Given $w \in \{-1,1\}^n$, we denote by $x \circ w$ the bitwise XOR (or in our case elementwise product, since we represent bits by $\{-1,1\}$) of x and w.

We also make use of basic notions from Boolean Fourier analysis. We use χ_S to denote the parity on $S \subseteq [n]$, $\chi_S(x) = (-1)^{\sum_{i \in S} x_i} = \prod_{i \in S} x_i$. The Fourier coefficients of a function $f : \{-1,1\}^n \to \mathbb{R}$ are denoted $\widehat{f}(S)$, with $\widehat{f}(S) = \langle f, \chi_S \rangle$, and $f = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S$.

5.2.1 Learning in the statistical query model

The statistical query (SQ) model was introduced by Kearns [Kea98] and has proved highly influential as a realistic learning model that also allows strong lower bounds; see [Fel16, Rey20] for surveys. Let D be a labeled distribution on $\{-1,1\}^n \times \{-1,1\}$. An SQ oracle for D is one that takes as input a query function $\phi : \{-1,1\}^n \times \{-1,1\} \to [-1,1]$ and a tolerance $\tau \in (0,1)$, and responds with a value in $[\mathbb{E}_D[\phi] - \tau, \mathbb{E}_D[\phi] + \tau]$. When a query is of the special form $\phi(x,y) = g(x)y$ for some $g : \{-1,1\}^n \to [-1,1]$, it is known as a correlational query, and is fully specified just by the function g. In the common case where D is actually $f(D)$ for some function $f : \{-1,1\}^n \to \{-1,1\}$ and distribution D on $\{-1,1\}^n$, the expectation of a correlational query is $\mathbb{E}_{(x,y) \sim f(D)}[g(x)y] = \mathbb{E}_{x \sim D}[f(x)g(x)] = \langle f, g \rangle_D$, and for this reason such queries are also called inner product queries.

We recall the definitions of (realizable) PAC and agnostic learning as applicable in the SQ model. Let $\mathcal{C}$ be a Boolean function class on $\{-1,1\}^n$. In the traditional, realizable PAC setting, the learner is given SQ access to $c(D)$ for an unknown $c \in \mathcal{C}$ and arbitrary distribution D on $\{-1,1\}^n$, and is said to learn $\mathcal{C}$ up to error ϵ if it is able to output a function h such that $\mathbb{P}_{x \sim D}[h(x) \neq c(x)] \leq \epsilon$. The learner is called a distribution-free learner if it has this guarantee irrespective of what D is. In the agnostic setting, the labels no longer arise from a function $c \in \mathcal{C}$. Instead the learner is given SQ access to an arbitrary labeled distribution D on $\{-1,1\}^n \times \{-1,1\}$, and the goal is to be competitive with the best-fitting classifier in $\mathcal{C}$. Letting $\mathsf{opt} = \inf_{c \in \mathcal{C}} \mathbb{P}_{(x,y) \sim D}[c(x) \neq y]$, a learner is said to agnostically learn $\mathcal{C}$ up to (excess) error ϵ if it is able to output a function h such that $\mathbb{P}_{(x,y) \sim D}[h(x) \neq y] \leq \mathsf{opt} + \epsilon$. Again, the learner is called distribution-free if this holds for all D.

For distribution-specific learners of Boolean classes, it is a well-known observation [BF02] that correlational SQs (CSQs) are equivalent to general SQs. This is not known to be the case in the distribution-free setting in which we operate, but it is known that a general SQ learner implies a CSQ learner when either sampling or nonuniformity is allowed [Fel08].

5.2.2 Correlational variance

In [MS22], the authors introduce a measure called the "variance" of a function class, or more precisely a family of labeled distributions, and use it to show lower bounds both on CSQ learning as well as approximation of this class. We use the term "correlational variance" to refer to this quantity for clarity. The setting is as follows. Let $\mathcal{A}$ be a family of function-distribution pairs (f, D), where $f : \{-1,1\}^n \to \{-1,1\}$ and D is a distribution on $\{-1,1\}^n$. $\mathcal{A}$ can also be seen as the family of the labeled distributions $f(D)$. Such a family will play exactly the same role as the notion of the "assumption class" from [KSS94]: namely, it is from an unknown member of this family that the learner receives labeled data (or SQ access to it).

Definition 5.2.1 (Correlational Variance)**.** Let $\mathcal{A}$ be a family of function-distribution pairs as above. For any $\phi : X \to [-1,1]$, define

$$\mathrm{Var}(\mathcal{A}, \phi) = \mathop{\mathbb{E}}_{(f,D)\ \mathcal{A}} \langle f, \phi \rangle_D^2 = \frac{1}{|\mathcal{A}|} \sum_{(f,D) \in \mathcal{A}} \langle f, \phi \rangle_D^2.$$

Then the "correlational variance" of $\mathcal{A}$ is defined as

$$\mathrm{Var}(\mathcal{A}) = \sup_{\|\phi\|_1 \le 1} \mathrm{Var}(\mathcal{A}, \phi).$$

In words, the correlational variance of a family captures the maximum variance (really the second moment) of the correlation of a query function ϕ with a random member of the family.

This measure yields lower bounds on SQ learning using correlational queries. The core of the proof is the following appealingly simple Markov/Chebyshev-style

lemma, which bounds the number of functions in a family that can be highly correlated with a query. We include it with a short proof so as to capture the essential function of the definition of correlational variance.

Lemma 5.2.2 ([MS22], Lemma 6)**.** *Let A be a family as above. Fix a query function $\phi : f \ 1, 1g^n \ ! \ f \ 1, 1g$ and a tolerance τ. Let A_ϕ denote the function-distribution pairs with correlation at least τ with ϕ, i.e. $f(f, D) \ 2 \ A \ j \ jhf, \phi i_D j \ \ \tau g$. Then $jA_\phi j \ \ \frac{\mathrm{Var}(A)}{\tau^2} jAj$.*

Proof. What fraction of pairs (f, D) in A satisfy $hf, \phi i_D^2 \ \ \tau^2$? By a simple Markov/Chebyshev bound, this fraction is at most $\frac{\mathrm{Var}(A,\phi)}{\tau^2} \ \ \frac{\mathrm{Var}(A)}{\tau^2}$. □

Roughly speaking, this means that an adversarial SQ oracle that responds with 0 to every query only allows the learner to rule out at most $\frac{\mathrm{Var}(A)}{\tau^2} jAj$ functions per query, essentially forcing even a weak learner to use $\Omega(\frac{\tau^2}{\mathrm{Var}(A)})$ queries in total. Formally, [MS22] show the following bound.

Theorem 5.2.3 ([MS22], Theorem 7)**.** *Let A be a family of function-distribution pairs as above. Consider a learner given CSQ access to an unknown $(f, D) \ 2 \ A$ with the goal of nding h such that the 0-1 error $\mathsf{P}_{x \ D}[f(x) \ 6= h(x)]$ is small. Then any such learner making only correlational queries of tolerance τ must use $\Omega(\frac{\tau^2}{\mathrm{Var}(A)})$ queries in order to output a function with 0-1 error better than $\frac{1}{2} \ \ \frac{\tau}{2}$. In particular, assuming $\tau \ \ \frac{1}{10}$ (say), $\Omega(\frac{\tau^2}{\mathrm{Var}(A)})$ queries are required in order to obtain 0-1 error at most $\frac{1}{3}$.*

The correlational variance of a family A can be bounded in terms of the spectral norm of the linear operator $M(A)$ mapping $\phi : X \ ! \ [\ 1, 1]$ to the vector $(hf, \phi i_D)_{(f,D) 2 A}$. Here the domain X will be $f \ 1, 1g^n$ for us. If we vectorize ϕ as $v(\phi) = (\phi(x))_{x 2 X}$, then $M(A)$ is the matrix of size $jAj \ \ jXj$ whose rows are indexed by $(f, D) \ 2 \ A$ and the columns by $x \ 2 \ X$, and whose entries are given by $[f(x)D(x)]_{(f,D) 2 A, x 2 X}$. We then have the following bound.

Lemma 5.2.4.

$$\mathrm{Var}(A) \quad \frac{jXj}{jAj}\, kM(A)k^2 .$$

Proof. This follows immediately from the observation that for any $\phi : X \;\; ! \;\; [\;\; 1, 1]$,

$$\mathrm{Var}(A, \phi) = \frac{1}{jAj}\, kM(A)v(\phi)k^2 \quad \frac{1}{jAj}\, kM(A)k^2\, kv(\phi)k^2 \quad \frac{jXj}{jAj}\, kM(A)k^2 ,$$

where $kv(\phi)k^2 \quad jXj$ since $\phi : X \;\; ! \;\; [\;\; 1, 1]$. □

Bounds on correlational variance also have implications for hardness of approximation, which we cover in Section 5.6.

5.2.3 Orthogonalizing distributions

The notion of an orthogonalizing distribution will be important to us, as it is the most general setting in which our results can be stated. It is a notion that has been used in many prior CSQ bounds and communication complexity results [She08a, She11, RS10, DKKZ20, DKZ20a].

Definition 5.2.5 (Orthogonalizing distribution)**.** Let $f : f \;\; 1, 1g^n \;\; ! \;\; f \;\; 1, 1g$ be a Boolean function. We say a distribution μ on $f \;\; 1, 1g^n$ orthogonalizes f up to degree d if for all polynomials of degree at most d, $hf, pi_\mu = 0$. In particular, $hf, \chi_S i_\mu = 0$ for all $jSj \quad d$.

For us a convenient way to use this definition will be to define a function $g = f\mu$, and to observe that for all $jSj \quad d$,

$$\widehat{g}(S) = hg, \chi_S i = 2 \;\; ^n \sum_{x2f \;\; 1,1g^n} f(x)\mu(x)\chi_S(x) = 2 \;\; ^n hf, \chi_S i_\mu = 0,$$

as well as that for all $S \quad [n]$, since $jhf, \chi_S i_\mu j \quad 1$,

$$j\widehat{g}(S)j = jhg, \chi_S ij = 2 \;\; ^n jhf, \chi_S i_\mu j \quad 2 \;\; ^n.$$

5.2.4 Threshold and approximate degree

Two of the most basic ways of using real polynomials to represent a Boolean function are to represent it as a polynomial threshold function, and to approximate it pointwise. These are both classical notions that have a long history in approximation theory, and both notions are accompanied by a beautiful duality theory. In both cases the dual characterization can be viewed in terms of orthogonalizing distributions.

Definition 5.2.6 (Threshold degree)**.** Let $f : f \quad 1, 1g^n \quad ! \quad f \quad 1, 1g$ be a function. The threshold degree of f, denoted deg (f), is the least degree of a real polynomial $p : X \quad ! \quad R$ such that $f(x) = \text{sign}(p(x))$ for all x.

It is a classical result that the threshold degree has a dual characterization in terms of an orthogonalizing distribution; see e.g. [She11, Theorem 3.3].

Theorem 5.2.7. *Let $f : f \quad 1, 1g^n \quad ! \quad f \quad 1, 1g$ be a function with threshold degree d. Then there exists a distribution μ on X that orthogonalizes f up to degree $d \quad 1$.*

Definition 5.2.8 (Approximate degree)**.** Let $f : f \quad 1, 1g^n \quad ! \quad f \quad 1, 1g$ be a function. The pointwise ϵ-approximate degree of f, denoted $\widetilde{\deg}_\epsilon(f)$, is the least degree of a real polynomial $p : f \quad 1, 1g^n \quad ! \quad R$ such that $jf(x) \quad p(x)j \quad \epsilon$ for all x.

Again, the following dual characterization is well-known; see e.g. [She11, Theorem 3.2].

Theorem 5.2.9. *Let $f : f \quad 1, 1g^n \quad ! \quad f \quad 1, 1g$ be a function with $\widetilde{\deg}_\epsilon(f) = d$. Then there exists a function $\psi : f \quad 1, 1g^n \quad ! \quad R$ such that:*

(a) $\sum_x \psi(x) f(x) > \epsilon$,

(b) $\sum_x j\psi(x)j = 1$,

(c) $h\psi, pi = 0$ *for all polynomials p of degree less than d.*

In particular, if we de ne a distribution $\mu(x) = j\psi(x)j$ and let $h(x) = \text{sign}(\psi(x))$, then we see that μ orthogonalizes h up to degree $d \quad 1$, and that $hh, f i_\mu > \epsilon$.

5.2.5 Pattern matrices

Our constructions of hard families make use of Sherstov's pattern matrix method [She11], along the lines of the construction of [MS22] for DNFs.

Definition 5.2.10 (Pattern matrix)**.** Let $f : \{-1,1\}^k \to \mathbb{R}$ be a function, and let $n > k$ be a multiple of k. Let $V(n,k)$ be the family of all size-k subsets $V \subseteq [n]$ of the following form: dividing $[n]$ into k consecutive blocks of size n/k, V consists of exactly one index from each block. (Notice that $|V(n,k)| = (n/k)^k$.) The (n,k,f)-pattern matrix is the matrix of size $2^n \times (n/k)^k 2^k$ whose rows are indexed by $x \in \{-1,1\}^n$ and columns by pairs $(V,w) \in V(n,k) \times \{-1,1\}^k$, and whose elements are given by $f(x_V - w)$.

The following definitions are similar but will make many of our theorems easier to state.

Definition 5.2.11 (Pattern matrix class)**.** Let $f : \{-1,1\}^k \to \mathbb{R}$ be a function, and let $n > k$ be a multiple of k. The (n,k,f)-pattern matrix class is the class of all functions on $\{-1,1\}^n$ of the form $x \mapsto f(x_V - w)$ for some $(V,w) \in V(n,k) \times \{-1,1\}^k$. We also sometimes use the term (n,k,f)-pattern matrix family for a family of function-distribution pairs where the functions involved are generated from f in this way.

Definition 5.2.12 (Pattern restrictions)**.** Let $C = \bigcup_{n>0} C_n$ be the union of some classes C_n of Boolean functions on $\{-1,1\}^n$. We say C is closed under pattern restrictions if for any k, n a multiple of k, and any $f \in C_k$, the function $x \mapsto f(x_V - w)$ on $\{-1,1\}^n$ lies in C_n for any $V \subseteq [n]$ of size k and $w \in \{-1,1\}^k$. In the common case where n is a small constant multiple of k, we will often be somewhat loose and not explicitly distinguish between C_k and C_n and just refer to C. Indeed, one can consider C_k to effectively be a subset of C_n using only some k out of n bits.

Most common Boolean classes are closed under pattern restrictions. The main notable exceptions are monotone classes such as monotone conjunctions.

We will need the following bound on the spectral norm of a pattern matrix in terms of the Fourier coefficients of f.

Theorem 5.2.13 ([She11], Theorem 4.3). *Let $f : f \quad 1,1g^k \quad ! \quad \mathsf{R}$, and let A be its (n,k,f)-pattern matrix. Let $s = 2^n 2^k (\frac{n}{k})^k$ be the number of entries in A. Then*

$$kAk = \frac{p}{s} \max_{S \quad [k]} \left\{ j\widehat{f}(S)j \left(\frac{k}{n}\right)^{jSj/2} \right\}.$$

5.3 Main theorem: correlational variance bounds via orthogonalizing distributions

Here we state our main theorem in its most general setting, that of orthogonalizing distributions. The theorem is a strong generalization of [MS22, Theorem 12], which proved such a result for the special case of DNF formulas, leveraging a powerful communication complexity result of Razborov and Sherstov [RS10]. Our main contribution lies in noting that this proof does not require the full strength of [RS10], and actually holds in a considerably general setting. Our lower bounds in terms of threshold and approximate degree follow as a result of this main theorem, since their dual characterizations furnish exactly the kinds of orthogonalizing distributions we need. The proof is a simple application of the pattern matrix method. It is worth noting that the functions in the hard family are all in fact juntas.

Theorem 5.3.1. *Let $n > k$ be a multiple of k. Let $f : f \quad 1,1g^k \quad ! \quad \mathsf{R}$ be a a Boolean function, and let μ be a distribution that orthogonalizes f up to degree $d \quad 1$, i.e. a distribution such that $hf, pi_\mu = 0$ for all polynomials of degree less than d. There exists a (n,k,f)-pattern matrix family A of function-distribution pairs on $f \quad 1,1g^n$ such that $\mathrm{Var}(A) \quad (\frac{n}{k}) \quad ^d$.*

Proof. For every $(V,w) \; 2 \; V(n,k) \quad f \quad 1,1g^k$, define $f_{V,w}(x) = f(x_V \quad w)$ and

$D_{V,w}(x) = 2^{k\ n}\mu(x_V \quad w)$. One can verify that $D_{V,w}$ is a valid distribution on $f \quad 1,1g^n$:

$$\begin{aligned}
\sum_{x2f\ \ 1,1g^n} D_{V,w}(x) &= 2^{k\ \ n} \sum_{x2f\ \ 1,1g^n} \mu(x_V \quad w) \\
&= 2^{k\ \ n} \sum_{z2f\ \ 1,1g^k} \sum_{\substack{x2f\ \ 1,1g^n: \\ x_V = z}} \mu(x_V \quad w) \\
&= 2^{k\ \ n} 2^{n\ \ k} \sum_{z2f\ \ 1,1g^k} \mu(z \quad w) \\
&= 1.
\end{aligned}$$

Our family will be

$$A = \left\{ (f_{V,w}, D_{V,w})\ j\ (V,w)\ 2\ V(n,k) \quad f \quad 1,1g^k \right\}.$$

To analyze this, let $g = 2^{k\ \ n} f\mu$, and observe that $M(A)$ is precisely the (transpose of the) (n,k,g)-pattern matrix. For all $jSj < d$, since $hf, \chi_S i_\mu = 0$, i.e. $\sum_x f(x)\chi_S(x)\mu(x) = 0$, we have that $\widehat{g}(S) = hg, \chi_S i = 0$. Moreover, we know that for all S,

$$\widehat{g}(S) = hg, \chi_S i = 2^{\ \ k} \sum_{x2f\ \ 1,1g^k} 2^{k\ \ n} f(x)\mu(x)\chi_S(x) \quad 2^{\ \ n} \sum_{x2f\ \ 1,1g^k} \mu(x) = 2^{\ \ n}.$$

Thus by Theorem 5.2.13, we have that

$$kM(A)k \quad p_{\overline{s}2^{\ \ n}} \left(\frac{k}{n}\right)^{d/2},$$

where $s = 2^n 2^k (n/k)^k$. Note that $jAj = 2^k (n/k)^k = 2^{\ \ n} s$. So finally by Lemma 5.2.4, we have

$$\mathrm{Var}(A) \quad \frac{2^n}{jAj} kM(A)k^2 \quad \frac{2^n}{2^{\ \ n} s} s2^{\ \ 2n} \left(\frac{k}{n}\right)^d = \left(\frac{k}{n}\right)^d.$$

□

The following corollary is immediate from Theorem 5.2.3.

Corollary 5.3.2. *Let A be the family de ned above. Any SQ learner for A making only correlational queries of tolerance $\tau < \frac{1}{10}$ must use $\Omega\left((\frac{n}{k})^d \tau^2\right)$ queries in order to output a function with 0-1 error at most $\frac{1}{3}$.*

5.4 CSQ lower bounds for PAC learning in terms of threshold degree

Our main lower bound for PAC learning in terms of threshold degree is the following.

Theorem 5.4.1. *Let $n > k$ be a multiple of k. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a function of threshold degree d. Let F denote the (n,k,f)-pattern matrix class. Any distribution-free SQ learner for F making only correlational queries of tolerance $\tau < \frac{1}{10}$ must use $\Omega\left(\left(\frac{n}{k}\right)^d \tau^2\right)$ queries in order to output a function with 0-1 error at most $\frac{1}{3}$.*

Proof. Letting μ be the orthogonalizing distribution (on $\{-1,1\}^k$) guaranteed by the dual characterization (Theorem 5.2.7), this theorem follows immediately by considering the family A from Theorem 5.3.1 and Corollary 5.3.2, since $\mathrm{Var}(A) \geq \left(\frac{n}{k}\right)^d$. □

We can now prove our original Theorem 5.1.1 as stated.

Proof of Theorem 5.1.1. Let C be a class closed under pattern restrictions (Definition 5.2.12), with threshold degree $\Omega(d)$. Pick a function $f : \{-1,1\}^{n/2} \to \{-1,1\}$ in C of threshold degree $\Omega(d)$. (Technically f lies in the version of C, call it $C_{n/2}$, on $n/2$ bits. But as noted in Definition 5.2.12, one can effectively view $C_{n/2}$ as being part of C. The threshold degree bound remains $\Omega(d)$ asymptotically.) Now the class F constructed above (Theorem 5.4.1, with $k = n/2$) must satisfy $F \subseteq C$. The theorem follows. □

One application of this theorem, already proved as a special case in [MS22], is a $2^{\Omega(n^{1/3})}$ SQ lower bound for learning DNF formulas on $\{-1,1\}^n$, proved by leveraging the well-known $\Omega(n^{1/3})$ Minksy–Papert threshold degree bound for DNFs [MP69]. This bound matches up to logarithmic factors in the exponent the algorithm of [KS04] for this problem.

Another application is to the circuit class AC^0, for which the following essentially-optimal threshold degree bound was recently shown.

Theorem 5.4.2 ([SW19]). *For any constant $\delta > 0$, there exists an AC^0 circuit on n bits with threshold degree $\Omega(n^{1-\delta})$.*

By Theorem 5.4.1, we obtain as a consequence a CSQ lower bound of $2^{-(n^{1-\delta})}$ (for any constant $\delta > 0$) for learning AC^0. This is essentially the strongest possible lower bound for distribution-free CSQ learning of AC^0.

Another important application is to intersections of two halfspaces on $\{-1,1\}^n$, for which Sherstov [She13] showed an optimal threshold degree bound of $\Omega(n)$. This yields a CSQ lower bound of $2^{-(n)}$ for distribution-free learning of this class, which is again the strongest possible and improves considerably on the bound of [KS07] of $2^{-(\sqrt{n})}$ for intersections of $\sqrt{n}$ halfspaces.

Needless to say, further applications hold for all classes with known threshold degree bounds, including symmetric functions, decision trees, and functions with known lower bounds on sensitivity [NS94, BDW02, Hua19, KSP20].

It is also worth noting that if we treat k as a free parameter, then Theorem 5.4.1 yields hard families of k-juntas. To our knowledge, the class of parities on k bits, i.e. $\{\chi_S \mid S \subseteq [n], |S| = k\}$, was the only class for which an optimal, exponential SQ lower bound of $\sum_{i=k}\binom{n}{i} = n^{-(k)}$ was known (under the uniform distribution), matching the algorithm of [MOS03] (itself only a polynomial improvement over brute force). We see that by picking $f : \{-1,1\}^k \to \{-1,1\}$ to be any function of threshold degree $\Omega(k)$, such as an intersection of two halfspaces [She13], then we can construct a family A of k-juntas with a nearly-optimal distribution-free CSQ lower bound of $(\frac{n}{k})^{-(k)}$.

5.5 CSQ lower bounds for agnostic learning in terms of approximate degree

Our lower bound for agnostic learning is obtained by exploiting the dual characterization of approximate degree (Theorem 5.2.9), which turns out to be precisely suited to this task. This is done as follows. Let f be a function with high $\frac{1}{2}$-approximate degree. First, since the dual function h is accompanied by an orthogonalizing distribution μ, we can use Theorem 5.3.1 to generate a family from h that is hard to learn. Next, since the dual function h and the original f are well-correlated, specifically $\langle f, h \rangle_\mu > \frac{1}{2}$, we can use an agnostic learner for a class generated from f to yield a (weak) PAC learner for the family generated from h. Since the latter problem is hard, the original problem is as well.

Lemma 5.5.1. *Let $n > k$ be a multiple of k. Let $f : \{-1,1\}^k \to \{-1,1\}$ be such that $\widetilde{\deg}_\epsilon(f) = d$, and let $h : \{-1,1\}^k \to \{-1,1\}$ and the orthogonalizing distribution μ on $\{-1,1\}^k$ be as given by its dual characterization (Theorem 5.2.9). There exists a (n,k,h)-pattern matrix family A of function-distribution pairs on $\{-1,1\}^n$ such that any SQ learner for A making only correlational queries of tolerance $\tau < \frac{1}{10}$ must use $\Omega\left(\left(\frac{n}{k}\right)^d \tau^2\right)$ queries in order to output a function with 0-1 error at most $\frac{1}{3}$.*

Proof. This is immediate from Theorem 5.3.1 and Corollary 5.3.2. □

This lets us prove an agnostic learning hardness result by reducing the problem of PAC learning A to agnostically learning a similar pattern matrix family generated from f.

Lemma 5.5.2. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be such that $\widetilde{\deg}_{1-\alpha}(f) = d$. Let F denote the (n,k,f)-pattern matrix class. Then a distribution-free agnostic learner for F capable of achieving 0-1 error $\mathrm{opt} + \epsilon$ yields a PAC learner capable of achieving 0-1 error $\alpha/2 + \epsilon$ for the family A in Lemma 5.5.1.*

Proof. For brevity let $D = D_{V,w}$ and let $f', h' = f_{V,w}, h_{V,w}$. It is not hard to see that $\langle f', h' \rangle_D = \langle f, h \rangle_\mu$:

$$\begin{aligned}
\langle f', h' \rangle_D &= \mathop{\mathbb{E}}_{x \sim D}[f'(x)h'(x)] \\
&= 2^{k-n} \sum_{x \in \{-1,1\}^n} f(x_V \circ w)h(x_V \circ w)\mu(x_V \circ w) \\
&= 2^{k-n} \sum_{z \in \{-1,1\}^k} \sum_{\substack{x \in \{-1,1\}^n: \\ x_V = z}} f(x_V \circ w)h(x_V \circ w)\mu(x_V \circ w) \\
&= 2^{k-n} 2^{n-k} \sum_{z \in \{-1,1\}^k} f(z \circ w)h(z \circ w)\mu(z \circ w) \\
&= \sum_{z \in \{-1,1\}^k} f(z)h(z)\mu(z) \\
&= \langle f, h \rangle_\mu.
\end{aligned}$$

So $\langle f', h' \rangle_D = \langle f, h \rangle_\mu > 1 - \alpha$, by Theorem 5.2.9. This means $\mathbb{P}_{x \sim D}[f'(x) \neq h'(x)] = (1 - \langle f', h' \rangle_D)/2 < \alpha/2$. In other words, each function-distribution pair $(h', D) \in A$ has $\mathrm{opt} = \inf_{f' \in F} \mathbb{P}_{x \sim D}[f'(x) \neq h'(x)] < \alpha/2$ with respect to F. So if we could agnostically learn F with error $\mathrm{opt} + \epsilon$ in a distribution-free way, then we could PAC-learn A with error $\alpha/2 + \epsilon$. □

Theorem 5.5.3. *Let $n > k$ be a multiple of k. Let $f : \{-1,1\}^k \to \{-1,1\}$ be such that $\widetilde{\deg}_{1/2}(f) = d$. Let F denote the (n,k,f)-pattern matrix class. Any distribution-free agnostic learner for F using only correlational statistical queries of tolerance $\tau < \frac{1}{10}$ requires at least $\Omega\left(\left(\frac{n}{k}\right)^d \tau^2\right)$ queries in order to output a function with excess 0-1 error $\frac{1}{100}$, i.e. true 0-1 error $\mathrm{opt} + \frac{1}{100}$.*

Proof. Suppose we had such a learner. By Lemma 5.5.2, taking $\alpha = \frac{1}{2}$, this would be a PAC learner for A capable of achieving 0-1 error at most $\frac{1}{4} + \frac{1}{100} < \frac{1}{3}$. Such a learner must obey the bound in Lemma 5.5.1. □

Theorem 5.1.2 now follows in just the same way as Theorem 5.1.1.

An important application of this theorem is the problem of agnostically learning conjunctions (likewise disjunctions) in the distribution-free setting. This was one of the original problems considered in [KSS94], and has seen hardness results in various restricted settings over the years, notable among which are a $2^{(\sqrt{n})}$ lower bound on Perceptron-based approaches (via the approximate rank) [KS10] and a super-polynomial (but not exponential) CSQ lower bound for learning monotone conjunctions under the uniform distribution [Fel12]. Since the $\frac{1}{2}$-approximate degree of conjunctions on n bits is $\Theta(\sqrt{n})$ [NS94, Pat92], we obtain a CSQ lower bound of $2^{(\sqrt{n})}$ for this problem. This is essentially the best possible CSQ lower bound for this problem.

Another important application is the problem of agnostically learning halfspaces [KKMS08]. Halfspaces can compute Majority functions on n bits, which have an approximate degree of $\Omega(n)$ [Pat92]. This yields a $2^{(n)}$ CSQ lower bound distribution-free agnostic learning of halfspaces, which is the strongest possible bound. Plugging in known approximate degree bounds for other classes such as symmetric Boolean functions [Pat92, BDW02] yield further applications.

One can also apply this theorem to the class AC^0, even though the resulting theorem is already implied by the PAC lower bound proved via threshold degree earlier. It is interesting that this bound can also be proved directly, via the following approximate degree bound.

Theorem 5.5.4 ([BT19]). *For any constant $\delta > 0$, there exists an AC^0 circuit on n bits with $\frac{1}{2}$-approximate degree $\Omega(n^{1-\delta})$.*

This yields a CSQ lower bound of $2^{(n^{1-\delta})}$ (for any constant δ) for distribution-free agnostic learning of AC^0.

5.5.1 CSQ lower bounds on attribute-efficient agnostic learning of sparse halfspaces

The problem of attribute-efficient learning [Blu90, BL97] formalizes a notion of learning in the presence of irrelevant attributes, and is an important open problem in learning theory. Consider the class of k-sparse halfspaces on $f \quad 1, 1g^n$. Since it has VC dimension $O(k \log n)$, from a statistical point of view a sample complexity of $\mathrm{poly}(k \log n)$ suffices to learn it (both in the PAC and agnostic settings). An efficient learner which achieves this sample complexity would be called an attribute-efficient learner. But despite years of research, no distribution-free attribute-efficient learners are known for this basic class.

In the SQ setting, the appropriate analog of sample complexity is the tolerance. Specifically, it takes $\Theta(1/\tau^2)$ samples to simulate a query of tolerance τ, and this is sometimes known as the estimation complexity of an SQ algorithm using tolerance τ. Accordingly, Feldman [Fel14] posed the following question as the problem of attribute-efficient SQ learning of sparse halfspaces: does there exist an SQ algorithm capable of learning k-sparse halfspaces on $f \quad 1, 1g^n$ only using $\mathrm{poly}(n)$ queries of tolerance $(k \log n) \ ^{O(1)}$? Though the question was originally asked in the PAC setting, here we answer it in the negative in the distribution-free, agnostic, CSQ setting. The result follows readily from our agnostic learning lower bound in terms of approximate degree.

Theorem 5.5.5. *There exists a family F of k-sparse halfspaces on $f \quad 1, 1g^n$ such that any distribution-free agnostic learner for F only using correlational queries of tolerance $\tau < \frac{1}{10}$ requires at least $(\frac{n}{k}) \ ^{(k)}\tau^2$ queries in order to agnostically learn F up to excess error $\frac{1}{10}$. In particular, for any $\tau = (k \log n) \ ^{O(1)}$, the lower bound remains $(\frac{n}{k}) \ ^{(k)}$ asymptotically, meaning no attribute-e cient CSQ learner exists.*

Proof. Taking $f : f \quad 1, 1g^k \ ! \ f \quad 1, 1g$ to be a halfspace with $\frac{1}{2}$-approximate degree $\Omega(k)$, such as Majority, this is a direct application of Theorem 5.5.3. □

5.6 Hardness of approximation in terms of approximate degree

In addition to CSQ lower bounds, [MS22] also prove hardness of approximation results for a family in terms of its correlational variance. Given a hypothesis class H and a family A of function-distribution pairs, a hardness of approximation result takes the form of a statement that in the worst case over a choice of (f, D) 2 A, no hypothesis h 2 H can even achieve nontrivial loss with respect to $f(D)$, i.e. approximate f wrt D. Naturally, this certainly implies that H cannot be used to learn A. Our bounds on correlational variance in terms of approximate degree have consequences for hardness of approximation by kernelized linear functions, establishing inherent limitations of kernel methods for learning A. In particular, they show that the polynomial kernel is essentially an optimal kernel map in the distribution-free setting. This amounts to a variant of a result already observed in [She11].

Let $f, h : f \quad 1, 1g^n \quad ! \quad R$ be functions, and let D be a distribution on $f \quad 1, 1g^n$. Let $L_{f(D)}(h)$ denote the squared loss $E_{x \quad D}[(h(x) \quad f(x))^2] = kh \quad fk_D^2$. Fix any embedding (or kernel feature map) $\psi : f \quad 1, 1g^n \quad ! \quad [\quad 1, 1]^N$ for any N, $B > 0$, and define the kernelized linear class

$$H_{\psi,B} = fx \; 7! \; \psi(x) \quad w \; j \; kwk_2 \quad Bg \, .$$

Then the following bound on approximation by $H_{\psi,B}$ holds.

Theorem 5.6.1 ([MS22]). *Let A be a family of function-distribution pairs on $f \quad 1, 1g^n$. Then*

$$\max_{(f,D)2A} \min_{h2H_{\psi,B}} L_{f(D)}(h) \quad E_{(f,D) \quad A} \min_{h2H_{\psi,B}} L_{f(D)}(h) \; > \; \frac{1}{2} \quad B \sqrt{p} \overline{N} \sqrt{\mathrm{Var}(A)}.$$

A similar but more involved result can also be proved for approximation by depth-two neural networks; see [MS22, Theorem 3].

We obtain the following consequence.

Theorem 5.6.2. *Let C be a Boolean function class closed under pattern restrictions, with $(1 \quad \alpha)$-approximate degree d, where α is a su ciently small constant ($\alpha = 1/16$ su ces). Let $H_{\psi,B}$ be the linear hypothesis class de ned above. Then there exists a family A of function-distribution pairs, with the functions lying in C, such that*

$$\max_{(f,D)2A} \min_{h2H_{\psi,B}} L_{f(D)}(h) \qquad \mathop{\mathsf{E}}_{(f,D) \ A} \min_{h2H_{\psi,B}} L_{f(D)}(h) \ > \ \frac{1}{8} \quad B^{\sqrt{N}2} \quad {}^{(d)}.$$

Proof. Let $f : \{1,1\}^{n/2} \ ! \ \{1,1\}$ be a function in C with $(1 \quad \alpha)$-approximate degree d. Let $g : \{1,1\}^{n/2} \ ! \ \{1,1\}$ and μ on $\{1,1\}^{n/2}$ be the accompanying dual function and distribution respectively, so that $\langle f, g \rangle_\mu \quad 1 \quad \alpha$.

Let A denote the $(n, n/2, f)$-pattern matrix family of function-distribution pairs arising from f and μ, i.e. pairs of the form $(f_{V,w}, D_{V,w})$, where $f_{V,w}(x) = f(x_V \quad w)$ and $D_{V,w}(x) = 2^{\ n/2}\mu(x_V \quad w)$. Similarly let B denote the $(n, n/2, g)$-pattern matrix arising from g and μ. Recall that $\mathrm{Var}(B) \quad 2^{\ d}$, since μ orthogonalizes g up to degree d.

Fix any V, w, and let $f' = f_{V,w}, g' = g_{V,w}, D' = D_{V,w}$ for brevity. It is easily calculated that $\langle f', g' \rangle_{D'} = \langle f, g \rangle_\mu \quad 1 \quad \alpha$, so that $\|f' \quad g'\|_{D'}^2 = 2 \quad 2\langle f', g' \rangle_{D'} \quad 2\alpha$. Now fix any $h \ 2 \ H_{\psi,B}$. We will argue that if h can approximate f' under D', then it can also approximate g', simply because f' and g' are close under D'. Indeed,

$$\begin{aligned} L_{g'(D')}(h) &= \|g' \quad h\|_{D'}^2 \\ &\quad 2\|f' \quad h\|_{D'}^2 + 2\|f' \quad g'\|_{D'}^2 \\ &\quad 2L_{f'(D')}(h) + 4\alpha, \end{aligned}$$

i.e.,

$$L_{f'(D')}(h) \quad \frac{1}{2} L_{g'(D')}(h) \quad 2\alpha.$$

Thus for every $(f', D') \ 2 \ A$, there is a corresponding $(g', D') \ 2 \ B$ such that

$$\min_{h2H_{\psi,B}} L_{f'(D')}(h) \quad \frac{1}{2} \min_{h2H_{\psi,B}} L_{g'(D')}(h) \quad 2\alpha,$$

and in fact this correspondence is one-to-one. So, taking the average,

$$\begin{aligned}\mathop{\mathsf{E}}_{(f^0,D^0)\ \ A}\ \min_{h2H_{\psi,B}}\ L_{f^0(D^0)}(h) \quad &\frac{1}{2}\mathop{\mathsf{E}}_{(g^0,D^0)\ \ B}\ \min_{h2H_{\psi,B}}\ L_{g^0(D^0)}(h) \quad 2\alpha \\ > \frac{1}{4} \quad &\frac{B}{2}\,\mathsf{p}\overline{N}\sqrt{\mathrm{Var}(B)} \quad 2\alpha \\ \frac{1}{8} \quad &\frac{B}{2}\,\mathsf{p}\overline{N2}^{\ d}\end{aligned}$$

for α 1/16. □

Thus we see that for any class C with approximate degree that is even superlogarithimic, i.e. $\omega(\log n)$, so that $2^{(d)}$ is negligible, i.e. $n^{\omega(1)}$, then no kernelized linear function (with B, N being poly(n)) can even weakly approximate C. If B is fixed, then the dimension N of the embedding must be taken to be $2^{(d)}$ for approximation to even be possible. This is essentially tight, since d is of course the approximate degree, and so approximation is certainly possible using the polynomial kernel up to degree d (so that $N = n^{O(d)}$). Thus in this sense the polynomial kernel is nearly optimal for distribution-free approximation.

5.7 A hard class of p-concepts under the uniform distribution

The correlational variance framework extends straightforwardly from Boolean functions to p-concepts. A p-concept c on $\mathsf{f}\ 1,1\mathsf{g}^n$ is a rule that to every $x\ \mathsf{2}\ \mathsf{f}\ 1,1\mathsf{g}^n$ assigns a random label $y\ \mathsf{2}\ \mathsf{f}\ 1,1\mathsf{g}$ such that $\mathsf{E}[y\mathsf{j}x] = c(x)$. Given a distribution D on $\mathsf{f}\ 1,1\mathsf{g}^n$, we will use $c(D)$ to denote the distribution on $\mathsf{f}\ 1,1\mathsf{g}^n \quad \mathsf{f}\ 1,1\mathsf{g}$ of labeled points (x,y) such that $x \quad D$ and $\mathsf{E}[y\mathsf{j}x] = c(x)$. The 0-1 loss of a function $h : \mathsf{f}\ 1,1\mathsf{g}^n\ \mathsf{!}\ \mathsf{f}\ 1,1\mathsf{g}$ with respect to a p-concept c over D is given by $\mathsf{P}_{(x,y)\ \ c(D)}[h(x)\ \mathsf{6}{=}\ y]$. Since the labels are random, in general no Boolean function can achieve 0 loss. The Boolean function that best fits c is given by sign(c), and it achieves loss

$$\mathop{\mathsf{P}}_{(x,y)\ \ c(D)}[\mathrm{sign}(c(x))\ \mathsf{6}{=}\ y] = \frac{1}{2}\ \ \frac{1}{2}\mathop{\mathsf{E}}_{(x,y)\ \ c(D)}[\mathrm{sign}(c(x))y] = \frac{1}{2}\ \ \frac{1}{2}\mathsf{h}c, \mathrm{sign}(c)\mathsf{i}_D = \frac{1}{2}\ \ \frac{1}{2}\mathsf{k}c\mathsf{k}_{1,D}\,,$$

where $\|c\|_{1,D} = \mathbb{E}_{x \in D}[|c(x)|]$. Thus the 1-norm $\|c\|_{1,D}$ characterizes the best achievable advantage over random guessing.

Let $\mathcal{A}$ be a family of pairs (c, D), where c is a p-concept and D is a distribution on $\{-1,1\}^n$. We will work in the distribution-specific setting where D is the same for all c, and is in fact just the uniform distribution U. In this case we know that CSQ is equivalent to general SQ. One can now define $\mathrm{Var}(\mathcal{A})$ exactly as before, and exactly the same proof as the one for Theorem 5.2.3 establishes the following theorem.

Theorem 5.7.1. *Let $\mathcal{A}$ be a family as above. Consider a learner given SQ access to an unknown $(c, D) \in \mathcal{A}$ with the goal of finding h such that the 0-1 error $\Pr_{x \in c(D)}[h(x) \neq y]$ is small. Then any such learner making only queries of tolerance τ must use $\Omega(\frac{\tau^2}{\mathrm{Var}(\mathcal{A})})$ queries in order to output a function with 0-1 error better than $\frac{1}{2} - \frac{\tau}{2}$.*

We can now prove a theorem very similar to Theorem 5.3.1, constructing a hard family of p-concepts from a function $f : \{-1,1\}^k \to \{-1,1\}$ and an orthogonalizing distribution μ. In essence, instead of considering pairs of the form (f, μ), we consider pairs of the form $(f\mu, U)$. Rather than consider $f\mu$ directly, it will be convenient to work with a scaling instead, so that we have greater control over the 1-norm. This requires an assumption on the dual distribution μ, namely that it satisfy a pointwise upper bound of the form $\mu(x) \leq \lambda 2^{-k}$ for all $x \in \{-1,1\}^k$. Notice that we may always naively take $\lambda = 2^k$. But in some cases it may be possible to pick λ smaller, which would allow our bounds to improve.

Theorem 5.7.2. *Let $n > k$ be a multiple of k. Let $f : \{-1,1\}^k \to \mathbb{R}$ be a a Boolean function, and let μ be a distribution that orthogonalizes f up to degree $d - 1$, i.e. a distribution such that $\langle f, p \rangle_\mu = 0$ for all polynomials of degree less than d. Assume that $\mu(x) \leq \lambda 2^{-k}$ for all $x \in \{-1,1\}^k$. Fix the underlying distribution on $\{-1,1\}^n$ to be uniform. There exists an $(n, k, f\mu)$-pattern matrix family $\mathcal{A}$ of p-concepts on $\{-1,1\}^n$ such that $\mathrm{Var}(\mathcal{A}) \geq \frac{1}{\lambda^2}(\frac{k}{n})^d$.*

Proof. Let U denote the uniform distribution on $\{-1,1\}^n$. For every $(V,w) \in V(n,k) \subseteq \{-1,1\}^k$, let $f_{V,w}(x) = f(x_V \circ w)$, $\mu_{V,w}(x) = \mu(x_V \circ w)$, and define $h_{V,w}(x) = \frac{2^k}{\lambda} f(x_V \circ w)\mu(x_V \circ w)$. Since $\mu \geq \lambda 2^{-k}$, $h_{V,w}$ is a well-defined p-concept, and its 1-norm wrt U is given by

$$\|h_{V,w}\|_{1,U} = 2^{-n} \sum_{x \in \{-1,1\}^n} \frac{2^k}{\lambda} \mu(x_V \circ w) = 2^{-n} \frac{2^k}{\lambda} 2^{n-k} = \frac{1}{\lambda}.$$

Its best-fitting Boolean function is $\mathrm{sign}(h_{V,w}) = f_{V,w}$. Our family will be

$$\mathcal{A} = \left\{ (h_{V,w}, U) \mid (V,w) \in V(n,k) \subseteq \{-1,1\}^k \right\}.$$

To analyze this, let $g = \frac{2^{k-n}}{\lambda} f\mu$, and observe that $M(\mathcal{A})$ is precisely the (transpose of the) (n,k,g)-pattern matrix. For all $|S| < d$, since $\langle f, \chi_S \rangle_\mu = 0$, i.e. $\sum_x f(x)\chi_S(x)\mu(x) = 0$, we have that $\widehat{g}(S) = \langle g, \chi_S \rangle = 0$. Moreover, we know that for all S,

$$\widehat{g}(S) = \langle g, \chi_S \rangle = 2^{-k} \sum_{x \in \{-1,1\}^k} \frac{2^{k-n}}{\lambda} f(x)\mu(x)\chi_S(x) \geq \frac{2^{-n}}{\lambda} \sum_{x \in \{-1,1\}^k} \mu(x) = \frac{2^{-n}}{\lambda}.$$

Thus by Theorem 5.2.13, we have that

$$\|M(\mathcal{A})\| \geq p\frac{2^{-n}}{s\lambda} \left(\frac{k}{n}\right)^{d/2},$$

where $s = 2^n 2^k (n/k)^k$. Note that $|\mathcal{A}| = 2^k (n/k)^k = 2^{-n} s$. So finally by Lemma 5.2.4, we have

$$\mathrm{Var}(\mathcal{A}) \geq \frac{2^n}{|\mathcal{A}|} \|M(\mathcal{A})\|^2 \geq \frac{2^n}{2^{-n} s} s \frac{2^{-2n}}{\lambda^2} \left(\frac{k}{n}\right)^d = \frac{1}{\lambda^2} \left(\frac{k}{n}\right)^d.$$

□

We can now prove hardness of agnostic learning in terms of approximate degree by showing that an agnostic learner could learn a hard family of p-concepts constructed as above. Specifically, let $f : \{-1,1\}^k \to \{-1,1\}$ be such that $\frac{1}{2}$-approximate degree is $\widetilde{\deg}_{1/2}(f) = d$ (we stress that this is a function of k, and perhaps best written

$d(k)$). Let $h : \{-1,1\}^k \to \{-1,1\}$ and the orthogonalizing distribution μ on $\{-1,1\}^k$ be as given by its dual characterization (Theorem 5.2.9). Assume that $\mu(x) \geq \lambda 2^{-k}$ for all $x \in \{-1,1\}^k$. Define $g_{V,w}(x) = \frac{2^k}{\lambda} h(x_V - w)\mu(x_V - w)$. Consider the p-concept family

$$\mathcal{A} = \left\{ (g_{V,w}, U) \mid (V,w) \in \mathcal{V}(n,k) \right\} \quad \{-1,1\}^k \to \{-1,1\} \Big\}.$$

By the previous theorems, we know that any learner for $\mathcal{A}$ using queries of tolerance τ requires $\Omega(\frac{\tau^2}{\mathrm{Var}(\mathcal{A})}) = \Omega(\left(\frac{n}{k}\right)^d \lambda^2\tau^2)$ queries in order to output a function with 0-1 error better than $\frac{1}{2} - \frac{\tau}{2}$.

Theorem 5.7.3. *Let $\mathcal{F}$ denote the (n,k,f)-pattern matrix class. Any agnostic learner for $\mathcal{F}$ under U using only statistical queries of tolerance $\tau < \frac{1}{8\lambda}$ requires at least $\Omega(\left(\frac{n}{k}\right)^d \lambda^2\tau^2)$ queries in order to output a function with excess 0-1 error $\frac{1}{8\lambda}$, i.e. true 0-1 error* $\mathrm{opt} + \frac{1}{8\lambda}$.

Proof. Suppose we had such a learner. Take the class of p-concepts $\mathcal{A}$ defined above, and consider any $g_{V,w}$ in $\mathcal{A}$. Let $g^0 = g_{V,w}$ for brevity. What is the optimal error $\mathrm{opt} = \inf_{f^0 \in \mathcal{F}} \mathbf{P}_{x \in U}[f^0(x) \neq g^0(x)]$ that any function in $\mathcal{F}$ achieves in approximating g^0 under U? This is the same as $\frac{1}{2} - \frac{1}{2}\sup_{f^0 \in \mathcal{F}} \langle f^0, g^0 \rangle_U$. Observe that $f^0 = f_{V,w}$ does quite well:

$$\begin{aligned}
\langle f^0, g^0 \rangle_U &= 2^{-n} \sum_{x \in \{-1,1\}^n} f(x_V - w)g(x_V - w) \\
&= \frac{2^k}{\lambda} 2^{-n} \sum_{x \in \{-1,1\}^n} f(x_V - w)h(x_V - w)\mu(x_V - w) \\
&= \frac{1}{\lambda} \sum_{z \in \{-1,1\}^k} f(z)h(z)\mu(z) \\
&= \frac{1}{\lambda} \langle f, h \rangle_\mu \\
&\leq \frac{1}{2\lambda},
\end{aligned}$$

since $\langle f, h \rangle_\mu \geq 1/2$ (by the dual characterization). Thus $\mathsf{opt} \geq \frac{1}{2} - \frac{1}{4\lambda}$. Now, if $\epsilon, \tau \leq \frac{1}{8\lambda}$, then our agnostic learner is able to achieve 0-1 error

$$\mathsf{opt} + \epsilon \leq \frac{1}{2} - \frac{1}{8\lambda} < \frac{1}{2} - \frac{\tau}{2}$$

when learning A. Therefore it must obey the $\Omega(\left(\frac{n}{k}\right)^d \lambda^2 \tau^2)$ lower bound on learning A. □

Overall, we see that the applicability of this result boils down to what λ we can actually achieve. We also know that this is actually a lower bound on uniform-distribution agnostic learning, so it can never improve on the [KKMS08] upper bound. We can trivially always take $\lambda = 2^k$, which corresponds to not scaling the p-concepts at all. We do, however, obtain some nontrivial results when we take $k = \Theta(\log n)$. For instance, for intersections of two halfspaces, we know that $d = \Theta(k)$ [She13], so the lower bound we get would be $(\frac{n}{\log n})^{(\log n)}$, i.e. quasipolynomial.

Chapter 6: A Moment-Matching Approach to Testable Learning and a New Characterization of Rademacher Complexity

6.1 Introduction

In the fundamental model of agnostic learning [KSS92, Vap98], a learner tries to output the best-fitting function from a concept class C with respect to an unknown labeled distribution D in the following sense: given sufficiently many labeled examples, with high probability it must produce a hypothesis with error at most $\mathsf{opt}(C, D) + \epsilon$ over D, where $\mathsf{opt}(C, D)$ denotes the optimal error achievable over D by any concept in C. No assumptions are made on the labels.

Agnostic learning is known to be computationally intractable for even the simplest function classes without making assumptions on the marginal [KSS92, KV94a, KS09, GR09, FGRW12, Dan16a, DSS16]. There is now a substantial literature of efficient agnostic learning algorithms under various distributional assumptions, the most common being that the marginal is Gaussian or Unif$\{-1,1\}^d$ (see e.g. [LMN93, BT96, KKMS08, KOS08, Kan11]). The problem of directly verifying this distributional assumption from samples, however, is often computationally infeasible (such as for Unif$\{-1,1\}^d$) or fundamentally ill-posed (as for $N(0, I_d)$[1]).

Since in the agnostic model we make no assumptions on the labels, we have no a priori estimate of $\mathsf{opt}(C, D)$, the error of the best-fitting classifier. Thus, a major (and often overlooked) issue with the agnostic learning model is that *it is unclear how to verify that the agnostic learner has actually succeeded.* Note that while we can

[1] To see why this is the case even when $d = 1$, for any finite sample size m, and consider a (random) discrete distribution $\widehat{D}$ that is uniform on $\omega(m^2)$ points drawn from $N(0,1)$. This distribution has TV distance 1 from $N(0,1)$ (since it is discrete), yet a sample of size m drawn from $\widehat{D}$ will have $o(1)$ TV distance from a sample of size m drawn directly from $N(0,1)$.

estimate the true error of the output hypothesis on a hold-out set (a.k.a. validation), we do not know its relationship to $\mathsf{opt}(C, D)$.

With this motivation in mind, very recent work of Rubinfeld and Vasilyan [RV23] introduced the elegant model of testable agnostic learning, or just testable learning for short. In this model, no assumptions are made on D, but there is a tester responsible for verifying whether the unknown marginal is suitably well-behaved. Whenever the tester accepts, the learner must succeed at producing a hypothesis with error at most $\mathsf{opt}(C, D) + \epsilon$ (with high probability). And to ensure nontriviality, whenever the unknown marginal is indeed a certain well-behaved target marginal D_X, the tester must accept (with high probability). We say the class C is testably learnable with respect to a target marginal D_X if there is a tester-learner pair meeting these conditions (see Definition 6.2.3).

In this model, [RV23] showed that halfspaces can be testably learned with respect to the standard Gaussian as well as the uniform distribution on the hypercube in time and sample complexity $d^{\widetilde{O}(1/\epsilon^4)}$. Their algorithm involves checking that the low-degree moments of the unknown marginal are close to those of the target marginal D_X. For the case where the target marginal is the standard Gaussian, they show that this implies concentration and anticoncentration properties of the unknown marginal and further prove that any distribution that satisfies such properties admits low-degree polynomial approximators for halfspaces. This analysis, however, is catered specifically to the case of halfspaces and the Gaussian target marginal. For the case where the target marginal is the uniform distribution on the hypercube, their proof combines the construction for the Gaussian case with the "critical index" framework of [DGJ+10].

6.1.1 Our results

Our main algorithmic contribution is a general framework that yields efficient testable learning algorithms for broad classes of functions and distributions (both

continuous and discrete). As we discuss in more detail below, our framework departs from the focus on constructing low-degree polynomial approximations with respect to absolute loss as in [RV23], which appears hard to extend to classes beyond a single halfspace. Instead, we rely on a new connection to a stronger type of approximator — sandwiching polynomials — that arises naturally in constructing pseudorandom generators for classes of Boolean functions.

As it turns out, many interesting and well-studied concept classes admit both sandwiching approximators and ordinary low-degree polynomial approximators of essentially the same degree, even though sandwiching is a formally stronger notion. As a result, we derive testable learning algorithms for halfspaces and more generally arbitrary functions of a bounded number of halfspaces with respect to any fixed strongly logconcave distribution. For the uniform distribution on the hypercube, we obtain algorithms for halfspaces, degree-2 PTFs, and constant-depth circuits. For each of these applications, our running times and sample complexity guarantees match the best known results for ordinary agnostic learning, thus showing that testable learning can often be achieved at no additional cost (see Theorem 6.4.1 and Theorem 6.5.2 for precise statements).

In particular, for the special case of testably learning a single halfspace with respect to the Gaussian or the uniform distribution on the hypercube, our results improve the $d^{\widetilde{O}(1/\epsilon^4)}$ running time and sample complexity guarantee shown in [RV23] to $d^{\widetilde{O}(1/\epsilon^2)}$, matching the best known (and conditionally optimal) results for ordinary agnostic learning. Moreover, our analysis extends to a broad family of distributions including strongly logconcave distributions.

We now describe our results and discuss our techniques in more detail.

Sandwiching polynomial approximation. Our starting point is a relationship between testable learning and a certain stronger notion of polynomial approximation that arises naturally in building pseudorandom generators for Boolean func-

tion classes. Specifically, a concept class C admits *sandwiching* approximations of degree k and error ϵ on D_X if for every function $f \in \mathsf{C}$, there exist two degree-k polynomials p_l, p_u such that *for every* x, $p_l(x) \leq f(x) \leq p_u(x)$, and moreover $\mathsf{E}_{D_X}[f - p_l], \mathsf{E}_{D_X}[p_u - f] \leq \epsilon$. Observe that this is a stronger requirement than the existence of approximating polynomials for C with respect to absolute loss, which only requires that for every f, there be a degree k polynomial p such that $\mathsf{E}_{D_X}[|p - f|] \leq \epsilon$.

The main theorem underlying our framework shows that unlike the existence of polynomial approximators with respect to absolute loss, the existence of sandwiching approximations universally translates into testable learning algorithms:

Theorem 6.1.1 (Testable learning using approximate moment matching; see Theorem 6.4.5)**.** *Let D_X be a distribution on X, and let C be a concept class mapping X to $\{\pm 1\}$. Let $k \in \mathbb{N}, \delta \in \mathbb{R}_+$ be degree and slack parameters, and let $\epsilon > 0$ be the error parameter. Suppose that every $f \in \mathsf{C}$ admits degree-k ϵ-sandwiching polynomials $p_l \leq f \leq p_u$ w.r.t. D_X such that $\|p_l\|_1, \|p_u\|_1 \leq \epsilon/\delta$, where $\|p_u\|_1$ (resp. $\|p_l\|_1$) refers to the ℓ_1 norm of the coefficients of p_u (resp. p_l). Suppose also that with high probability over a sample of size $d^{O(k)}$, the empirical moments of degree at most k of D_X are within δ of their true moments. Then C can be testably learned w.r.t. D_X up to excess error $O(\epsilon)$ in sample and time complexity $d^{O(k)}/\operatorname{poly}(\epsilon)$.*

Theorem 6.1.1 relies on a simple tester: verify that the empirical moments of degree at most k are close enough to the those of D_X. The correctness of the tester relies on the claim that sandwiching polynomials for C under D_X are also sandwiching polynomials for C under the uniform distribution $\widehat{D}$ on a large enough sample from D_X, with an additional error that scales proportional to the ℓ_1 norm of the coefficients of the polynomial approximators. Thus, whenever we have sandwiching approximators with appropriate bounds on the ℓ_1 norm of the coefficient vectors, our testable learner can simply use the now-standard degree-k (absolute-loss) polynomial regression algorithm of [KKMS08].

The work of [KKMS08] showed that the existence of low-degree (not necessarily sandwiching) polynomial approximators with respect to absolute loss suffices for *ordinary* agnostic learning. Our theorem on sandwiching polynomial approximators can be thought of as the natural counterpart to their condition but for *testable* agnostic learning.

We stress that merely having ordinary polynomial approximators for C with respect to absolute loss under D_X — as opposed to sandwiching polynomials — would not have sufficed for Theorem 6.1.1. This is because such approximators do not readily translate to approximators for C under a different distribution D' that only approximately matches low-degree moments with D_X (i.e., passes our test for D_X). The crucial role played by sandwiching approximation is that it allows such a transfer principle: sandwiching approximators for C under D_X (with sufficiently small coefficients) are also sandwiching approximators for C under such a D' (see Corollary 6.3.3).

Our main task now reduces to constructing sandwiching polynomials with sufficiently small coefficients. Since proofs of existence of sandwiching polynomials are sometimes nonconstructive (e.g., for constant-depth circuits over the hypercube, where the existence of such polynomials follows from LP duality [Baz09]), we require new techniques to prove bounds on the ℓ_1 norm of the coefficients. We make progress by crucially exploiting a form of *approximate* duality between sandwiching polynomials and moment matching.

Moment matching and sandwiching polynomials. The duality between fooling using moment matching and the existence of sandwiching polynomials is well-known in the setting of the Boolean hypercube [Baz09], where moment matching $\mathrm{Unif}\{-1,1\}^d$ up to degree k is equivalent to k-wise independence. We need an approximate version of this duality in order to derive a bound on the coefficients of the approximating polynomials. Moreover, we need duality to hold over continuous

domains for our applications to non-discrete settings (such as Gaussian and strongly logconcave distributions). A duality relating *exact* moment matching and sandwiching approximations over general domains was proved in [KM13].

We derive the following general duality result, which tells us that approximate moment matching fools a class C over D_X iff every concept in C admits a pair of sandwiching polynomials with sufficiently small coefficients. Our proof relies on establishing a strong duality result for a certain *semi-in nite* linear program using tools from general conic duality [Sha01].

Theorem 6.1.2 (Fooling using approximate moment matching ( ) sandwiching approximation; see Theorem 6.3.2)**.** *Let D_X be a distribution on X, and let C be a concept class mapping X to f 1g. Let k 2 N, δ 2 R$_+$ be degree and slack parameters, and let $\epsilon > 0$ be the error parameter. The following are equivalent:*

- *(Approximate moment matching fools C.) For all f 2 C and for all distributions D^0 whose moments of degree at most k are within δ of those of D_X, we have j E$_{D^0}$[f] E$_{D_X}$[f]j ϵ.*
- *(Existence of sandwiching polynomials with bounded coe cients for C.) For all f 2 C, there exist degree-k polynomials p_l, p_u such that p_l f p_u (pointwise over R^d), and*

$$\mathop{\mathsf{E}}_{D_X}[p_u \quad f] + \delta k p_u k_1 \quad \epsilon, \qquad \mathop{\mathsf{E}}_{D_X}[f \quad p_l] + \delta k p_l k_1 \quad \epsilon,$$

where kp_uk$_1$ (resp. kp_lk$_1$) refers to the ℓ_1 norm of the coe cients of p_u (resp. p_l).

Applications: testably learning functions of halfspaces and more. Combining Theorems 6.1.1 and 6.1.2, we obtain a clean framework for testable learning that reduces the task to establishing that approximate low-degree moment matching fools the target concept class over the target marginal. As our main application, we

show that any function of a constant number of halfspaces over R^d can be testably learned up to excess error ϵ in sample and time complexity $d^{\widetilde{O}(1/\epsilon^2)}$ with respect to any distribution whose directional projections are sufficiently anticoncentrated and have strictly sub-exponentially decaying tails:

Definition 6.1.3. We say a distribution D_X on R^d is anticoncentrated and has α-strictly subexponential tails if the following hold:

(a) *α-strictly subexponential tails:* For all $\|u\| = 1$, $\mathsf{P}[|\langle x, u\rangle| > t]$ $\exp(-Ct^{1+\alpha})$ for some constant C.

(b) *Anticoncentration:* For all $\|u\| = 1$ and continuous intervals T R, we have $\mathsf{P}[\langle x, u\rangle \in T]$ $C'|T|$ for some constant C'.

Theorem 6.1.4 (Testably learning functions of halfspaces; see Theorem 6.5.2)**.** *Let C be the class of functions of a constant number of halfspaces over R^d. Let D_X be a distribution that is anticoncentrated and has α-strictly subexponential tails (Definition 6.1.3). Then C can be testably learned w.r.t. D_X up to excess error ϵ using sample and time complexity $d^{\widetilde{O}(\epsilon^{-(1+\alpha)/\alpha})}$.*

We note that even for ordinary agnostic learning, the above result is an exponential improvement in the dependence on ϵ in the degree of the sandwiching polynomial compared to prior constructions of [KM13].

On the flip side, note that even though our framework handles D_X that come from a fairly broad family, our tester must know the low-degree moments of the particular D_X with respect to which it is expected to succeed. This is true for the approach of [RV23] as well, and it is an interesting open question whether this can be relaxed.

The class of distributions that are anticoncentrated and have strictly subexponential tails is fairly general and includes Gaussians, the uniform distribution on the unit sphere, and more generally, any strongly logconcave distribution [SW14]

(and in fact all of these examples have $\alpha = 1$). This latter class includes the uniform distribution on any convex body with smooth boundary [BE85] and in particular, additive Gaussian smoothening of any convex body. Theorem 6.1.4 already matches the upper bound of [KKMS08] as well as known statistical-query (SQ) [GGK20, DKZ20a, DKPZ21] and cryptographic [DKR23, Tie23] lower bounds for ordinary agnostic learning of a single halfspace with respect to the Gaussian distribution. It also generalizes and improves the main algorithmic result of [RV23], who showed such a result for a single halfspace with time and sample complexity $d^{\widetilde{O}(1/\epsilon^4)}$.

The key technical result underlying Theorem 6.1.4 is a proof that any distribution that approximately matches degree-$\widetilde{O}(\epsilon^{(1+\alpha)/\alpha})$ moments with a distribution D_X which is anticoncentrated and has α-strictly subexponential tails fools functions of halfspaces with respect to D_X (see Theorem 6.5.6). Similar to the approach of [KM13], we rely on powerful methods arising from the classical theory of moments [KR96] and metric distances in probability [Zol84, RKSF13] to show that whenever the moments of D_X are strictly sub-exponential, moment closeness implies distribution closeness in the so-called λ-metric (see Section 6.5.1).

We also apply our framework to immediately obtain testable learning results with respect to the Uniform$_{\{0,1\}^d}$ in time $d^{O(k)}$ for classes $\mathcal{C}$ that are fooled by k-wise independence, including halfspaces [DGJ+10], degree-2 PTFs [DKN10], and constant-depth circuits [Bra10], with running time and sample complexity that matches their ordinary agnostic counterparts; see Theorem 6.4.1. Over the hypercube, the fact that approximate moment matching — i.e. almost k-wise independence — suffices to fool such classes is immediate by a result due to [AGM03].

Moments vs anticoncentration. Theorem 6.1.4 immediately implies that one can test anticoncentration properties of all directional marginals of a broad family of distributions by checking only the low-degree moments.

Corollary 6.1.5 (Anticoncentration from approximate moment matching; see Corollary 6.5.7)**.** *Fix $\epsilon > 0$ and a distribution D_X that is anticoncentrated and has α-strictly subexponential tails. Let D' be any distribution whose moments of degree at most $k = \widetilde{O}(\epsilon^{(1+\alpha)/\alpha})$ match those of D_X up to an additive slack of $d^{\widetilde{O}(k)}$. Then for any $\|u\| = 1$ and any continuous interval T R, $\mathsf{P}_{x \sim D'}[\langle x, u\rangle \in T] \leq \mathsf{P}_{x \sim D_X}[\langle x, u\rangle \in T] + \epsilon$.*

This statement relates anticoncentration phenomena to structure in low-degree moments. In particular, any distribution that matches the first degree-$\widetilde{O}(1/\epsilon^2)$ moments of a strongly logconcave distribution must have all its directional marginals anticoncentrated up to an additive error of ϵ. In addition to being a basic result in probability, such a connection relates to verifying anticoncentration of all directional marginals from a small sample. Finding verification subroutines that extend beyond Gaussian (and the uniform distribution on the sphere) have a host of applications in algorithmic robust statistics and immediately yield efficient robust algorithms for list-decodable linear regression [KKK19, RY20a] and covariance estimation [BK21, RY20b, IK22], and robust clustering [BK20, BDH+20] of mixtures for broader families of distributions than currently known.

For the specific case of Gaussian distributions (and the uniform distribution on the unit d-dimensional sphere), such a property for the case when T is an origin centered interval was first proved in a sequence of works that introduced *certifiable anticoncentration* in the context of algorithmic robust statistics [KKK19, RY20a]. Their proofs use a polynomial approximator for the "box function" (see e.g. [KKK19, Appendix A]) and show that degree-$\widetilde{O}(1/\epsilon^2)$ moments are enough to ensure ϵ-approximate anticoncentration for origin centered intervals T. A similar argument based on approximations for the box function was used by [RV23] to show that matching degree-$\widetilde{O}(1/\epsilon^4)$ moments of Gaussian implies ϵ-approximate anticoncentration for all intervals T as above. This quartic dependence in the order of moments required appears necessary in a proof that constructs polynomial approximations for

the box function. Our argument above circumvents this bottlneck in these previous techniques and recovers the ϵ-additive error anticoncentration from matching just the degree-$\widetilde{O}(1/\epsilon^2)$ moments.

Comparison to the algorithmic technique of [RV23]. As their main algorithmic result, Rubinfeld and Vasilyan [RV23] gave a testable learning algorithm for halfspaces that uses $d^{O(1/\epsilon^4)}$ time and samples. At a high level, the chief difference between our approach and theirs is that theirs relies on explicitly constructing polynomial approximators for any distribution D that matches low-degree moments with the target D_X, whereas ours uses a "black-box" transfer principle (Corollary 6.3.3) showing that sandwiching approximators under D_X immediately yield sandwiching approximators under D.

In more detail, their algorithm uses the fact that halfspaces admit a low-degree polynomial approximator with respect to a distribution D whenever D is anticoncentrated and has subgaussian low-degree moments. In order to verify that the empirical distribution on a large enough Gaussian sample possesses these two properties, they relate anticoncentration to low-degree moments via polynomial approximators for the box function as described above. For the case where the target marginal is the uniform distribution on the hypercube, they reuse their approximator for the box function and show that it yields a polynomial approximator for *regular* halfspaces with respect to almost k-wise independent distributions. They then utilize the "critical index" framework of [DGJ+10] to reduce the case of general halfspaces to the regular case.

Such an approach is possible for the simple setting of halfspaces over the Gaussian or the uniform distribution on the hypercube (with a suboptimal quartic dependence on $1/\epsilon$), but it is already unclear how to extend it to halfspaces on non-product distributions (where Fourier methods fail) or to more expressive concept classes such as functions of halfspaces.

In contrast, an appeal to sandwiching approximation allows us to extend our testable learning results to non-anticoncentrated discrete distributions such as the uniform distribution on the hypercube, more expressive concept classes such as constant depth circuits on the hypercube and functions of halfspaces on continuous distributions, and to a broad family of distributions including all strongly logconcave distributions.

Sample complexity of testable learning and Rademacher complexity. One of our main contributions is a complete characterization of the sample complexity of testable learning. Similar to how VC-dimension corresponds to the sample complexity of distribution-free agnostic learning, we show that Rademacher complexity is the key quantity that controls the sample complexity of testable learning. Recall that the Rademacher complexity of a class C w.r.t. D_X at sample size m is given by

$$R_m(C, D_X) = \mathop{\mathbb{E}}_{\{x_i\}_{i\in[m]} \sim D_X^m} \mathop{\mathbb{E}}_{\sigma \sim \{\pm 1\}^m} \sup_{f\in C} \left| \frac{1}{m} \sum_{i\in[m]} \sigma_i f(x_i) \right|.$$

This measure plays an important role in statistical learning theory since it controls the uniform convergence of empirical losses to true losses over all $f \in C$ (see Theorem 6.2.8). To our knowledge, our result is the first natural *model-based characterization* of Rademacher complexity. We obtain precise upper and lower bounds on the sample complexity of testable learning within excess error ϵ purely in terms of Rademacher complexity:

Theorem 6.1.6 (Rademacher complexity characterizes testable learning, see Theorems 6.6.1 and 6.6.2)**.** *Let D_X be a distribution on X, let C be a concept class mapping X to $\{\pm 1\}$, and let $\epsilon > 0$ be the error parameter.*

- *(Upper bound.) Let m be such that $R_m(C, D_X) \le \epsilon/5$. Then C can be testably learned w.r.t. D_X up to excess error ϵ using sample complexity $m + O(1/\epsilon^2)$.*

- *(Lower bound.) Let M be such that $R_M(C, D_X)$ 5ϵ, and assume M $\Theta(1/\epsilon^2)$. Then the sample complexity required to testably learn C w.r.t. D_X up to excess error ϵ is at least $\Omega(\sqrt{M})$.*

This characterization yields an interesting separation between ordinary distribution-specific agnostic learning and testable learning. For the former, while uniform convergence is always a sufficient condition, it is not necessary, as witnessed by examples such as convex sets in Gaussian space [KOS08] and monotone Boolean functions [BT96] (see Section 6.6.2.1). Indeed, the sample complexity of distribution-specific agnostic learning is known to be characterized by the $L^1(D_X)$ metric entropy rather than the Rademacher complexity [BI91]. In contrast, it is the Rademacher complexity that provides the right characterization for testable learning. Thus, testable learning is a natural supervised learning model for which bounded Rademacher complexity, and hence uniform convergence, provides a *necessary and su cient* condition for learning. For further discussion, see Section 6.7.1.

6.1.2 Subsequent work and open questions

Subsequent to the initial appearance in conference proceedings of both this work [GKK23] and that of Rubinfeld and Vasilyan [RV23], there have been a few exciting developments addressing some natural questions left open by these works. One such natural question arises from the fact that the tester-learners for halfspaces in these works inherently incur an expensive $d^{\mathsf{poly}(1/\epsilon)}$ dependence on d and $1/\epsilon$, and we know that such a dependence is necessary if we seek the "strong" agnostic learning guarantee of $\mathsf{opt} + \epsilon$ (even in the ordinary, non-testable setting) [GGK20, DKZ20a, DKPZ21, DKR23, Tie23]. Can we obtain more efficient algorithms by relaxing the optimality guarantee we seek? The work of [GKSV23a, DKK+23] shows that the answer is yes: for the case where the target marginal is Gaussian, one can in fact design fully polynomial-time tester-learners for origin-centered halfspaces with the weaker agnostic guarantee of $O(\mathsf{opt})$. Interestingly, both these works make use of the

key structural pseudorandomness result of this chapter (Theorem 6.5.6).

An even more natural and compelling question concerns the key role played by the target marginal D_X in the current framework (Definition 6.2.3). As mentioned earlier, the moment-matching tests in this work (and in fact all tester-learners discussed thus far) are inherently tailored to the particular D_X with respect to which the tester-learner is required to succeed; indeed, these tests would reject most distributions that are far from D_X, however well-behaved. Can we design tester-learners that succeed simultaneously with respect to a large class of structured distributions? This question was answered affirmatively by the very recent work of [GKSV23b], which builds on and subsumes [GKSV23a]. They show fully polynomial-time tester-learners for origin-centered halfspaces that guarantee error $O(\mathsf{opt})$ and accept whenever the marginal is *any* distribution that satisfies a Poincaré inequality (and some mild anticoncentration properties). This latter class of distributions includes all strongly log-concave distributions. In fact, under the Kannan–Lóvasz–Simonovits [KLS95] conjecture, it includes all log-concave distributions.

Interestingly, the natural "strong agnostic learning" version of the aforementioned result remains open. That is, can we design tester-learners for halfspaces that guarantee error $\mathsf{opt}+\epsilon$ (even at the cost of $d^{\mathsf{poly}(1/\epsilon)}$ time) while accepting a wide class $\mathcal{T}$ of structured distributions? One approach that would suffice (by the results in this chapter) is to design a tester that checks whether the low-degree empirical moment tensor matches *any* of the low-degree moment tensors of distributions in $\mathcal{T}$. For $\mathcal{T}$ being say the class of all (strongly) log-concave distributions, this is an interesting and open algorithmic question.

Many other important questions remain open. From the statistical point of view, the current characterization in terms of Rademacher complexity has a quadratic gap between the upper and lower bounds; can we provide a characterization that is tight up to constant factors? More generally, the model of testable learning is still very new, and interesting directions include designing tester-learners for richer concept

classes, as well as extending the types of guarantees that one provides testers for in the first place (error optimality guarantees are just one such type). Its connections to other areas of computational learning theory that are motivated by considerations of verifiability, such as classification with abstention [BW08] and reliable agnostic learning [KKM12], also remain to be fully explored.

6.1.3 Other related work

In the broader context of trustworthy or verifiable machine learning, there have been a number of works studying various different formulations of verifiability. These include: classification with abstention (see e.g. [BW08, CDM16]), in which a classifier is allowed to abstain from producing a prediction for certain points; reliable agnostic learning [KKM12], in which one seeks to control both the error as well as (say) the false positive rate (and which is closely related to a form of transductive learning [GKKM20, KK21]); and cryptographic interactive proofs for machine learning [GRSY21], in which one seeks to design an interactive proof protocol to verify the error guarantee claimed by a (potentially malicious) learner. Another very recent work that tackles issues raised by unrealistic distributional assumptions is that of [BLMT23], which proposes learning algorithms whose running times naturally scale with a measure of complexity of the underlying (unknown and arbitrary) distribution.

In pseudorandomness, the duality between fooling using bounded independence and sandwiching approximation is a fundamental tool for showing that k-wise independence fools various classes [Baz09, Bra10, DGJ+10]. Its more general statement in terms of moment matching was observed by [KM13] (see also [KKM13]), who used it to obtain low-degree sandwiching polynomials for functions of halfspaces w.r.t. logconcave distributions (their constructions do not give any insight on the ℓ_1 norm of the coefficients). We build on their approach for our main application, namely testably learning functions of halfspaces with respect to Gaussians, and obtain exponentially improved degree bounds in terms of ϵ along with effective bounds on the size of the coefficients.

In statistical learning theory and nonparametric regression, one of the basic objectives is to place tight bounds on the excess risk $L(\widehat{f})$ $\inf_{f \in \mathcal{C}} L(f)$ and on the generalization gap $|\widehat{L}_m(\widehat{f})$ $L(\widehat{f})|$ of an estimator $\widehat{f}$ in various settings (see e.g. [BM97, VdG00, Tsy08]). In particular, there is a long line of work studying data-dependent bounds on these quantities in terms of measures such as the Rademacher complexity and various refinements and variants thereof [KP00, Kol01, BBL02, BM02, BBM05, Kol06]. Our sample complexity upper bound applies a simple such data-dependent bound to the testable learning setting. In terms of lower bounds, [KR21] have studied bounds on the minimal error of any ERM estimator, in the additive Gaussian noise setting, in terms of the Gaussian complexity. None of these works, however, consider a model directly comparable to testable learning.

Statistical characterizations of PAC learning have also been well-studied. In the distribution-free setting, it is very well-known that the sample complexity is characterized fully by the VC-dimension, and equivalent to uniform convergence [Vap98]. For distribution-specific agnostic setting, [BI91] obtained a characterization in terms of the $L^1(D_X)$ metric entropy, i.e. the log covering number w.r.t. the metric $\rho(f, g) = \mathsf{E}_{D_X}[|f$ $g|]$. Work by [KL18] (see also [Vad17]) proposed a characterization of efficient agnostic learning using the so-called refutation complexity, and interpreted it as a computational analog of Rademacher complexity. Results of [SSSS10] showed that in Vapnik's General Setting of Learning, the sample complexity is in general characterized by notions of algorithmic stability rather than uniform convergence. In modern deep learning theory, the failures of the uniform convergence paradigm in the overparameterized regime have been much studied (see e.g. [ZBH+21, NK19]); we refer the reader to [BMR21, Bel21] for surveys.

6.2 Preliminaries

6.2.1 Notation and conventions

We denote the domain by X, which for us is always either $\mathbb{R}^d$ or $\{-1,1\}^d$, and labels always lie in $\{-1,1\}$. We use C to denote a concept class mapping X to $\{-1,1\}$. We use D_X to denote a well-behaved distribution on X (i.e. the target marginal, such as $N(0, I_d)$ or $\mathrm{Unif}\{-1,1\}^d$), and we use the calligraphic $\mathcal{D}$ to denote labeled distributions on $X \times \{-1,1\}$. We denote a size-m (labeled) sample drawn from $\mathcal{D}$ by $S \sim \mathcal{D}^m$. If $S = \{(x_i, y_i)\}_{i \in [m]}$, then we use $S_X = \{x_i\}_{i \in [m]}$ to denote its "marginal", i.e. the unlabeled sample.

Our loss function throughout will be the 0-1 loss function, $\ell(\widehat{y}, y) = \mathbb{1}[\widehat{y} \neq y]$. Given a labeled distribution $\mathcal{D}$, we denote the population loss by $L(f, \mathcal{D}) = \mathbb{P}_{(x,y) \sim \mathcal{D}}[f(x) \neq y]$ (or just $L(f)$ when $\mathcal{D}$ is implicit), and the empirical loss on a size-m sample $S \sim \mathcal{D}^m$ by $\widehat{L}_m(f, S) = \mathbb{P}_{(x_i,y_i) \sim S}[f(x_i) \neq y_i]$ (or just $\widehat{L}_m(f)$ when S is implicit). We follow the convention of denoting empirical quantities using a hat and a subscript to denote the sample size (as in $\widehat{L}_m$). We use $\mathsf{opt}(C, \mathcal{D})$ to denote $\inf_{f \in C} L(f, \mathcal{D})$.

We follow the following conventions when working with monomials over X. For any multi-index $I \in \mathbb{N}^d$, let $|I| = \sum_j I_j$ denote its degree (or sometimes order), and let x_I denote the monomial $\prod_{j \in [d]} x_j^{I_j}$. We use $\mathcal{I}(k, d) = \{I \in \mathbb{N}^d : |I| \leq k\}$ to denote the set of multi-indices of degree at most k. For a vector $\Delta \in \mathbb{R}_+^{|\mathcal{I}(k,d)|}$ and a degree-k polynomial $p : \mathbb{R}^d \to \mathbb{R}$ given by $p(x) = \sum_{I \in \mathcal{I}(k,d)} p_I x_I$, we use $\|\Delta, |p|\|_1$ to denote $\sum_{I \in \mathcal{I}(k,d)} |p_I| \Delta_I$. This may be thought of as the Δ-weighted ℓ_1 norm of the coefficients of p.

We use $a \approx b$ and $a \lesssim b$ to denote equalities and inequalities up to constants. It will be convenient to state Stirling's approximation in the following form [Rob55]: for all $n \geq 1$, $n! \approx e \cdot n^n n^{1/2}$. We will also make use of the double factorial $n!! = n(n-2)(n-4)\cdots$, satisfying $n! = n!!(n-1)!!$ for even n.

Throughout this chapter, we use the term "with high probability" to mean "with probability at least 0.99" (or any other sufficiently large constant) for simplicity. In all cases, confidence may be amplified using standard repetition arguments.

6.2.2 Learning models

Here we formally define the learning models we work with. Let D_X be a distribution on X, where X is either R^d or $f \quad 1g^d$, and let C be a concept class mapping X to $f \quad 1g$.

Definition 6.2.1 (Distribution-specific agnostic learning)**.** We say a learner A agnostically learns C w.r.t. D_X up to excess error ϵ if for any D on $X \quad f \quad 1g$ with marginal D_X, given sufficiently many examples drawn from D, with high probability A outputs a hypothesis h such that $L(h) \quad \mathsf{opt}(C, D) + \epsilon$. Here, recall that $L(f) = \mathsf{P}_{(x,y) \quad D}[f(x) \not= y]$ and $\mathsf{opt}(C, D) = \inf_{f 2 C} L(f)$.

We recall the standard result of [KKMS08] that shows that polynomial approximators with respect to the absolute loss yield agnostic learning algorithms.

Theorem 6.2.2 ([KKMS08])**.** *Suppose that for every f 2 C, there exists a degree-k polynomial $p : X \quad ! \quad \mathsf{R}$ such that $\mathsf{E}_{x \quad D_X}[jf(x) \quad p(x)j] \quad \epsilon$. Then there exists a simple agnostic learner (based on degree-k polynomial regression w.r.t. the absolute loss) for learning C w.r.t. D_X up to excess error ϵ in time and sample complexity $d^{O(k)}/\operatorname{poly}(\epsilon)$.*

We now formally define testable learning.

Definition 6.2.3 (Testable agnostic learning, [RV23])**.** We say a tester-learner pair (T, A) testably learns C w.r.t. D_X up to excess error ϵ if for any distribution D on $X \quad f \quad 1g$, the following conditions are met:

- (Soundness/composability.) If D is such that the tester T accepts with high probability over a sample drawn from D, then the learner A succeeds in agnostically learning C w.r.t. D, i.e. with high probability it produces a hypothesis h such that $L(h) \quad \mathsf{opt}(C, D) + \epsilon$.

- (Completeness.) Whenever $\mathcal{D}$ truly has marginal D_X on X, the tester T accepts with high probability.

Again, here "with high probability" may be taken to be "with probability at least 0.99" for simplicity, and the confidence in each step may be amplified using standard repetition arguments.

Note that as stated, it is not strictly necessary for the tester T and the learner A to work with the same sample, and the definition may be interpreted as saying "if T accepts $\mathcal{D}$ (w.h.p.), then A must succeed over $\mathcal{D}$ (w.h.p.)". The algorithms and characterizations we give in this chapter have a stronger "data-dependent" guarantee, where both T and A operate on the same sample S drawn from $\mathcal{D}$, and have the following interpretation: "if T accepts S, then A must succeed over S (as well as generalize to $\mathcal{D}$ w.h.p.)".

6.2.3 Bounded independence and sandwiching polynomials over the hypercube

In this section, let U denote Unif$\{-1,1\}^d$.

Definition 6.2.4. We say a distribution D on $\{-1,1\}^d$ is (δ,k)-independent if for all $|I| \leq k$, $|\mathbb{E}_D[x_I]| \leq \delta$. When $\delta = 0$, we simply call D a k-wise independent distribution.

We say that a concept class $\mathcal{C}$ is ϵ-fooled by (δ,k)-independence (resp. k-wise independence) if for every $f \in \mathcal{C}$ and any (δ,k)-independent (resp. k-wise independent) D, $|\mathbb{E}_D[f] - \mathbb{E}_U[f]| \leq \epsilon$.

Notice that saying D is (δ,k)-independent is exactly equivalent to saying that the moments of degree at most k of D are within δ of those of U (which, of course, are all 0).

We now recall a fundamental duality result in pseudorandomness, which states that bounded independence fools a class $\mathcal{C}$ iff it admits sandwiching polynomials w.r.t. U.

Theorem 6.2.5 ([Baz09, Thm A.1]). *Let $f : \{0,1\}^d \to \{0,1\}$, let $\epsilon > 0$ be the error parameter, and let $k \in \mathbb{N}, \delta > 0$ be the degree and slack parameters. The following are equivalent:*

(a) (δ, k)-independence fools f.

(b) There exist degree-k polynomials p_l, p_u such that $p_l \le f \le p_u$ (pointwise over $\{0,1\}^d$), and

$$\mathop{\mathbb{E}}_{U}[p_u - f] + \delta \|p_u\|_1 \le \epsilon, \qquad \mathop{\mathbb{E}}_{U}[f - p_l] + \delta \|p_l\|_1 \le \epsilon,$$

where for a polynomial $p(x) = \sum_I p_I x_I$ we use $\|p\|_1$ to denote $\sum_{I \neq 0} |p_I|$, i.e. the ℓ_1 norm of its (nonconstant) coefficients.

The following theorem, showing that a (δ, k)-independent distribution is statistically close to being k-wise independent, will also be useful to us.

Theorem 6.2.6 ([AGM03, Thm 2.1]). *Let D be a (δ, k)-independent distribution on $\{0,1\}^d$. Then there exists a k-wise independent distribution D' that has TV distance at most δd^k from D.*

6.2.4 Rademacher complexity

The Rademacher complexity is one of the most well-studied measures of the complexity of a function class in statistical learning theory, and may be intuitively thought of as measuring the ability of a function class to fit a randomly-labeled sample. The following definitions and theorems are now standard in the literature (see e.g. [BBL03, BM02, Bar14, BMR21] and references therein).

Definition 6.2.7 (Rademacher complexity)**.** Consider a sample of m points $S_X = \{x_i\}_{i \in [m]} \sim D_X^m$. The empirical Rademacher complexity of the class C w.r.t. this sample is defined to be

$$\widehat{R}_m(C, S_X) = \mathbb{E}_{\sigma \sim \{\pm 1\}^m} \sup_{f \in C} \left| \frac{1}{m} \sum_{i \in [m]} \sigma_i f(x_i) \right|. \tag{6.2.1}$$

Note that this is a random variable depending on S_X. The (expected) Rademacher complexity of C w.r.t. D_X at sample size m is defined to be

$$R_m(C, D_X) = \mathbb{E}_{S_X \sim D_X^m} \widehat{R}_m(C, S_X). \tag{6.2.2}$$

Sometimes we simply say $R_m(C)$ (resp. $\widehat{R}_m(C)$) when D_X (resp. S_X) is clear from context.

The next theorem states that the Rademacher complexity of a class tightly controls uniform convergence, i.e. bounds on the quantity $\sup_{f \in C} |L(f) - \widehat{L}_m(f)|$, where L and $\widehat{L}_m$ are the population and empirical loss functionals. The upper bound here follows from a so-called symmetrization argument, while the lower bound follows from a desymmetrization argument. In our statement, we specialize to the case of the 0-1 loss.[2]

Theorem 6.2.8 (see e.g. [Bar14])**.** *Let C be a class of functions mapping X to $\{\pm 1\}$. Let D be a distribution on $X \times \{\pm 1\}$ with marginal D_X on X, and let $S \sim D^m$ be a random sample of size m drawn from D. For any $f \in C$, let $L(f) = \mathbb{P}_{(x,y) \sim D}[f(x) \neq y]$, and let $\widehat{L}_m(f) = \mathbb{P}_{(x_i,y_i) \sim S}[f(x_i) \neq y_i]$. Then with probability $1 - \delta$ over the draw of S, we have*

$$\frac{1}{4} R_m(C) - \Theta\left(\sqrt{\frac{\log(1/\delta)}{m}}\right) \leq \sup_{f \in C} \left| L(f) - \widehat{L}_m(f) \right| \leq R_m(C) + \Theta\left(\sqrt{\frac{\log(1/\delta)}{m}}\right).$$

[2] For general loss functions ℓ, one would define a "loss class" $\ell \circ C = \{(x, y) \mapsto \ell(f(x), y) \mid f \in C\}$ and state such a result in terms of $R_m(\ell \circ C)$. In the case of the 0-1 loss function, it is known that $R_m(\ell \circ C) = \frac{1}{2} R_m(C)$.

The following useful facts characterize the concentration of the quantities defining the Rademacher complexity and follow by standard applications of McDiarmid's inequality. Assume that the range of C is bounded in $[\ \ 1, 1]$.

The first fact is that the empirical Rademacher complexity $\widehat{R}_m(C)$ concentrates tightly around the expected Rademacher complexity $R_m(C)$. Formally, with probability at least $1\ \ \delta$ over a sample $S_X = \{x_i\}_{i \in [m]}\ \ D_X{}^m$ of size m, we have

$$\left|R_m(C, D_X)\ \ \widehat{R}_m(C, S_X)\right|\ \ O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right). \tag{6.2.3}$$

The second is that for any fixed sample S of size m, the random variable $\sup_{f \in C} \frac{1}{m}\sum_i \sigma_i f(x_i)$ concentrates tightly around its expectation, $\widehat{R}_m(C)$. Formally, with probability at least $1\ \ \delta$ over $\sigma\ \ \{1\}^m$ we have

$$\left|\widehat{R}_m(C, S_X)\ \ \sup_{f \in C} \frac{1}{m}\sum_i \sigma_i f(x_i)\right|\ \ O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right). \tag{6.2.4}$$

Combining Theorem 6.2.8 with Eq. (6.2.3), we actually have the following data-dependent generalization guarantee for a sample S in terms of $\widehat{R}_m(C, S_X)$ itself: with probability at least $1\ \ \delta$ over the draw of S, for every $f \in C$,

$$\left|L(f)\ \ \widehat{L}_m(f)\right|\ \ \widehat{R}_m(C) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right). \tag{6.2.5}$$

6.3 Duality

In this section we state the duality between fooling using approximate moment matching and sandwiching polynomials. This is a generalization of duality over the hypercube, Theorem 6.2.5, to continuous domains and more general distributions. Note that a version of duality over $\mathbb{R}^d$, albeit only for exact moment matching, was stated in [KM13, Lemma 3.3].

Definition 6.3.1 (Approximate moment matching)**.** Let $k \in \mathbb{N}$ be a degree parameter, and let $\Delta \in \mathbb{R}_+^{\binom{k+d}{d}}$ be a slack parameter, satisfying $\Delta_0 := \Delta_{(0,\ldots,0)} = 0$ and

$\Delta_I > 0$ for all other $I \in I(k,d)$. We say that two distributions D, D' on X match moments of degree (or order) at most k up to slack Δ if $|\mathbb{E}_D[x_I] - \mathbb{E}_{D'}[x_I]| \leq \Delta_I$ for all $I \in I(k,d)$.

The reason for allowing the slack Δ_I to depend on I is that in general we expect the scale of the moments to vary widely with I (as with the Gaussian, for example). The empty index $I_0 = 0 = (0, \ldots, 0)$ plays a special role, since $x_{I_0} = 1$ and $\mathbb{E}_D[1] = 1$ for any valid distribution, meaning we may assume $\Delta_0 = 0$ without loss of generality.

We can now state the main theorem. We prove this theorem using conic LP duality [Sha01], taking care to establish strong duality, but the essential argument is similar to Bazzi's proof of Theorem 6.2.5.

Theorem 6.3.2 (Duality). *Let $k \in \mathbb{N}, \Delta \in \mathbb{R}_+^{|I(k,d)|}$ be the degree and slack parameters, as in Definition 6.3.1. Let $f : X \to \mathbb{R}$ be a function, and let D be a distribution on X. The following are equivalent:*

(a) (Approximate moment matching fools f w.r.t. D.) For any distribution D' whose moments up to order k match those of D up to Δ, we have $|\mathbb{E}_D[f] - \mathbb{E}_{D'}[f]| \leq \epsilon$.

(b) (Existence of sandwiching polynomials with bounded coefficients for f w.r.t. D.) There exist degree-k polynomials p_l, p_u such that $p_l \leq f \leq p_u$ (pointwise over $\mathbb{R}^d$), and

$$\mathop{\mathbb{E}}_{D}[p_u - f] + \langle \Delta, |p_u| \rangle \leq \epsilon, \qquad \mathop{\mathbb{E}}_{D}[f - p_l] + \langle \Delta, |p_l| \rangle \leq \epsilon.$$

(Recall that for a degree-k polynomial $p(x) = \sum_I p_I x_I$, we use $\langle \Delta, |p| \rangle$ to denote $\sum_{I \in I(k,d)} |p_I| \Delta_I$.)

Proof. Let $\sigma_I = \mathbb{E}_D[x_I]$. Let P_d be the set of all Borel probability measures on $\mathbb{R}^d$. Consider the following semi-infinite linear program, which seeks to maximize $\mathbb{E}_{D'}[f]$

over all probability distributions D' on $\mathbb{R}^d$ that approximately match moments with D:

$$\sup_{D' \in P_d} \mathop{\mathbb{E}}_{D'}[f] \tag{6.3.1}$$

$$\text{subject to} \qquad \sigma_I \quad \Delta_I \quad \mathop{\mathbb{E}}_{D'}[x_I] \quad \sigma_I + \Delta_I \quad \forall I \in I(k,d) \tag{6.3.2}$$

The case of $I = (0, \ldots, 0)$ is special: here $\sigma_0 = \mathbb{E}_D[1] = 1$ and $\Delta_0 = 0$, so the corresponding constraint becomes simply $\mathbb{E}_{D'}[1] = 1$, which is equivalent to requiring that D' be a valid probability measure.

The dual LP turns out to be equivalent to the following, with variable $\beta \in \mathbb{R}^{|I(k,d)|}$:

$$\inf_{\beta \in \mathbb{R}^{|I(k,d)|+1}} \sum_{I \in I(k,d)} \beta_I \sigma_I + \sum_{I \in I(k,d)} |\beta_I| \Delta_I \tag{6.3.3}$$

$$\text{subject to} \qquad \sum_{I \in I(k,d)} \beta_I x_I \quad f(x) \quad \forall x \in \mathbb{R}^d \tag{6.3.4}$$

Notice that the primal LP (Eq. (6.3.1)) is feasible (indeed, by $D' = D$), and moreover, we claim that strong duality holds. Accepting this for a moment, denote the common optimum of Eqs. (6.3.1) and (6.3.3) by γ. The claim that approximate moment matching fools f (in a one-sided fashion) w.r.t. D is the same as asserting $\gamma \quad \mathbb{E}_D[f] + \epsilon$. Take β to be an optimal solution to the dual, and let $p_u(x) = \sum_{I \in I(k,d)} \beta_I x_I$. (In fact, this correspondence between degree-k polynomials and their coefficient vectors allows us to equivalently view the dual as optimizing over such polynomials instead of their coefficients.) The dual then tells us that $p_u \quad f$ pointwise, and

$$\gamma = \sum_{I \in I(k,d)} \beta_I \sigma_I + \sum_{I \in I(k,d)} |\beta_I| \Delta_I = \mathop{\mathbb{E}}_{D}[p_u] + \langle \Delta, |p_u| \rangle \quad \mathop{\mathbb{E}}_{D}[f] + \epsilon,$$

establishing the existence of the upper sandwiching polynomial. To obtain the lower sandwiching polynomial, we replace the objective of the primal with $\mathbb{E}_{D'}[f]$ and repeat the same argument, this time using the fact that the common optimum γ' satisfies $\gamma' \quad \mathbb{E}_D[f] + \epsilon$ (i.e., effectively replacing f with $\quad f$ throughout). This establishes the desired equivalence if we accept strong duality.

Formally, it remains to properly justify that the primal LP (Eq. (6.3.1)) is well-posed, that Eq. (6.3.3) is indeed the dual of Eq. (6.3.1), and that strong duality holds. We do so in Section D.1 using results from general conic LP duality [Sha01]. □

To see how Theorem 6.1.2 may be recovered from this, simply set Δ to be δ in every coordinate.

For the purposes of testing using moment matching, one can only ever hope to check that the unknown marginal (D', say) *approximately* matches moments with the target marginal. Approximate duality — and specifically the appearance of the quantities $\langle \Delta, |p_u| \rangle$ and $\langle \Delta, |p_l| \rangle$ — turns out to be precisely what we need to guarantee sandwiching polynomials even w.r.t. such a D'. This is our black-box transfer principle.

Corollary 6.3.3 (Transfer principle)**.** *Let $f, D, k, \Delta, \epsilon$ satisfy the conditions of Theorem 6.3.2, and let $p_l \quad f \quad p_u$ be the resulting sandwiching polynomials for f w.r.t. D. Consider any particular D' whose moments up to order k match those of D up to Δ. Then p_l, p_u are sandwiching polynomials for f w.r.t. D' as well, satisfying*

$$\mathop{\mathbb{E}}_{D'}[p_u \quad f] \quad 2\epsilon, \qquad \mathop{\mathbb{E}}_{D'}[f \quad p_l] \quad 2\epsilon.$$

Proof. By the first part of Theorem 6.3.2, we know $|\mathbb{E}_D[f] \quad \mathbb{E}_{D'}[f]| \quad \epsilon$. Thus

$$\left|\mathop{\mathbb{E}}_{D}[p_u \quad f] \quad \mathop{\mathbb{E}}_{D'}[p_u \quad f]\right| \quad \left|\mathop{\mathbb{E}}_{D}[p_u] \quad \mathop{\mathbb{E}}_{D'}[p_u]\right| + \left|\mathop{\mathbb{E}}_{D}[f] \quad \mathop{\mathbb{E}}_{D'}[f]\right| \tag{6.3.5}$$

$$\langle \Delta, |p_u| \rangle + \epsilon. \tag{6.3.6}$$

Applying the second part of Theorem 6.3.2, this means

$$\mathop{\mathbb{E}}_{D'}[p_u \quad f] \quad \mathop{\mathbb{E}}_{D}[p_u \quad f] + \langle \Delta, |p_u| \rangle + \epsilon \tag{6.3.7}$$

$$2\epsilon. \tag{6.3.8}$$

The argument for p_l is exactly the same. □

6.4 Testable learning via moment matching

6.4.1 Warm-up: testable learning over the hypercube via k-wise independence

The main ideas of our approach are already illustrated in the setting of the Boolean hypercube. A key technical ingredient for us will be the fact that an almost k-wise independent distribution is statistically close to being truly k-wise independent (Theorem 6.2.6).

Theorem 6.4.1. *Let C be any concept class that is $\frac{\epsilon}{4}$-fooled by k-wise independence. Then C can be testably learned w.r.t.* Unif$\{0,1\}^d$ *up to excess error ϵ with time and sample complexity $d^{O(k)}/\epsilon^2$.*

Proof. Let the unknown labeled distribution be $\mathcal{D}$. Let $S \sim \mathcal{D}^m$ be the labeled sample given to (T, A) (where the sample size m will be picked later), and let S_X be its (unlabeled) marginal. Let $\widehat{D}_m$ be the induced empirical distribution, i.e. the uniform distribution over S_X.

The tester T and algorithm A are simple: the tester checks that the empirical moments (or biases) up to degree k are all no larger than $\delta = \epsilon d^{-k}/4$ in magnitude (i.e. that the empirical distribution is (δ, k)-independent), and the algorithm runs degree-k polynomial regression (Theorem 6.2.2) over the sample.

It is clear that when $\mathcal{D}$ indeed has marginal exactly Unif$\{0,1\}^d$ (or indeed any $(\delta/2, k)$-independent distribution), then by taking $m = d^k/\delta^2 = d^{(k)}/\epsilon^2$ sufficiently large, we can ensure with high probability all the empirical moments of order at most k concentrate about their true moments up to δ (by a standard Hoeffding plus union bound). That is, with high probability $\widehat{D}_m$ will indeed be (δ, k)-independent, and the tester will accept. This verifies completeness.

To verify soundness, suppose that $\widehat{D}_m$ is indeed (δ, k)-independent. By Theorem 6.2.6, this means that $\widehat{D}_m$ has TV distance at most $\delta d^k = \epsilon/4$ from a truly

k-wise independent distribution. This in turn means that $\widehat{D}_m$ (and indeed any (δ, k)-independent distribution) $\frac{\epsilon}{2}$-fools C. We now appeal to duality, stated here in some generality as Theorem 6.3.2, although in the setting of the hypercube this theorem reduces exactly to the form in Theorem 6.2.5. Formally, observe that for every $f \in \mathsf{C}$, condition (a) of Theorem 6.3.2 is satisfied (with $D = \mathrm{Unif}_{\{0,1\}^d}$, and where the slack parameter Δ is now simply δ in every coordinate). This allows us to apply Corollary 6.3.3 to conclude that there exist ϵ-sandwiching polynomials for C w.r.t. $\widehat{D}_m$. By Theorem 6.2.2, this ensures the learner succeeds at learning C up to error $\mathsf{opt}(\mathsf{C}, D) + \epsilon$ with high probability. This proves the theorem. □

We may apply this theorem to obtain testable learning w.r.t. $\mathrm{Unif}_{\{0,1\}^d}$ for halfspaces, degree-2 PTFs, and constant-depth circuits.

Corollary 6.4.2. *Let C be the class of halfspaces over $\{0,1\}^d$. Let $\epsilon > 0$, and let $k = \widetilde{O}(1/\epsilon^2)$. Then C is ϵ-fooled by k-wise independence [DGJ+10], and hence it can be testably learned w.r.t. $\mathrm{Unif}_{\{0,1\}^d}$ up to excess error ϵ with time and sample complexity $d^{O(k)}$.*

Corollary 6.4.3. *Let C be the class of degree-2 polynomial threshold functions over $\{0,1\}^d$. Let $\epsilon > 0$, and let $k = \widetilde{O}(1/\epsilon^9)$. Then C is ϵ-fooled by k-wise independence [DKN10], and hence it can be testably learned w.r.t. $\mathrm{Unif}_{\{0,1\}^d}$ up to excess error ϵ with time and sample complexity $d^{O(k)}$.*

Corollary 6.4.4. *Let C be the class of depth-t AC^0 circuits of size s over $\{0,1\}^d$. Let $\epsilon > 0$, and let $k = (\log s)^{O(t)} \log(1/\epsilon)$. Then C is ϵ-fooled by k-wise independence [Bra10, Tal17, HS19], and hence it can be testably learned w.r.t. $\mathrm{Unif}_{\{0,1\}^d}$ up to excess error ϵ with time and sample complexity $d^{O(k)}$.*

6.4.2 A general algorithm using moment matching

We now give a more general algorithm for testable learning that does not need the target distribution to be k-wise independent. In this case, our tester will

check that the low-degree moments of the empirical distribution are close to those of the target distribution. The correctness of our tester is a consequence of duality (Theorem 6.3.2).

Theorem 6.4.5. *Let D_X be a distribution on X, and let C be a concept class mapping X to $\{\pm 1\}$. Let $k \in \mathbb{N}, \Delta \in \mathbb{R}_+^{|I|(k,d)|}$ be the degree and slack parameters, as in Definition 6.3.1, and let $\epsilon > 0$ be the error parameter. Suppose the following conditions hold:*

(a) (Empirical moments concentrate around true moments.) There exists m large enough that with high probability over a sample $S_X \sim D_X{}^m$, the corresponding empirical distribution $\widehat{D}_m$ matches moments of degree at most k with D_X up to slack Δ.

(b) (Existence of sandwiching polynomials with bounded coefficients for C, or equivalently approximate moment matching fools C.) For every $f \in C$, there exist degree-k sandwiching polynomials $p_l \le f \le p_u$ such that

$$\mathop{\mathbb{E}}_{D_X}[p_u - f] + \langle \Delta, |p_u| \rangle \le \frac{\epsilon}{2}, \qquad \mathop{\mathbb{E}}_{D_X}[f - p_l] + \langle \Delta, |p_l| \rangle \le \frac{\epsilon}{2}.$$

(Recall that for a degree-k polynomial $p(x) = \sum_I p_I x_I$, we use $\langle \Delta, |p| \rangle$ to denote $\sum_{I \in |(k,d)|} |p_I| \Delta_I$.)

Equivalently, for every $f \in C$ and for any distribution D' whose moments up to order k match those of D_X up to Δ, we have $|\mathbb{E}_D[f] - \mathbb{E}_{D'}[f]| \le \frac{\epsilon}{2}$.

Then C can be testably learned w.r.t. D_X up to excess error ϵ using time and sample complexity $m + d^{O(k)}$. Moreover, the tester T and learner A are simple: T tests whether the empirical moments up to order k match those of D_X up to Δ, and A performs degree-k polynomial regression over the sample (Theorem 6.2.2).

Proof. Let the unknown labeled distribution be D, and let $S \sim D^m$ be the sample given to (T, A). First we verify completeness. By assumption, when D indeed has

marginal D_X, then m is large enough that with high probability over S, the empirical moments concentrate about the true moments up to Δ, and hence T accepts.

As for soundness, suppose that T accepts, i.e. that the empirical distribution $\widehat{D}_m$ indeed matches order-k moments with D_X up to Δ. Observe that our condition (b) is the same as condition (b) of Theorem 6.3.2 is satisfied. Thus we may apply Corollary 6.3.3 (with $D = D_X$ and $D^{\emptyset} = \widehat{D}_m$) to conclude that there exist degree-k ϵ-sandwiching polynomials for C w.r.t. $\widehat{D}_m$. By Theorem 6.2.2, we have that degree-k polynomial regression achieves error $\mathsf{opt}(C, D) + \epsilon$ with high probability. (This implicitly assumes that the degree-k polynomial fitting S will generalize to D, which will be true by classic VC theory whenever $m \quad d^{O(k)}/\epsilon^2$ since the VC dimension of degree-k polynomials (with bounded coefficients, as here) is at most $d^{O(k)}$. If this is not the case, we may replace m with $m + d^{O(k)}/\epsilon^2$.) □

To see how Theorem 6.1.1 may be recovered from this, simply set Δ to be δ in every coordinate, and also rescale ϵ appropriately.

6.5 Testably learning functions of halfspaces over strictly subexponential distributions

In this section we apply Theorem 6.4.5 to prove that we can testably learn functions of halfspaces w.r.t. a target marginal D_X on $X = \mathsf{R}^d$ that is anticoncentrated and has strictly subexponential tails in the sense of Definition 6.1.3 from the introduction.

Definition 6.5.1 (Restatement of Definition 6.1.3)**.** We say a distribution D_X on R^d is anticoncentrated and has α-strictly subexponential tails if the following hold:

(a) For all $\|u\|_2 = 1$, $\mathsf{P}[|\langle x, u\rangle| > t] \quad \exp(\ C_1 t^{1+\alpha})$ for some constant C_1.

(b) For all $\|u\|_2 = 1$ and $k \in \mathsf{N}$, $\mathsf{E}[|\langle x, u\rangle|^k]^{1/k} \quad C_2 k^{1/(1+\alpha)}$ for some constant C_2.

(c) For all $\|u\|_2 = 1$ and continuous intervals $T \subseteq \mathbb{R}$, we have $\mathbb{P}[\langle x, u\rangle \in T] \leq C_3|T|$ for some constant C_3.

The first two conditions are a strengthening of the usual definition of subexponential distributions (see e.g. [Ver18]), and standard arguments show that the two are actually equivalent. The third asks directional marginals of D_X to be anticoncentrated. Examples include all *strongly* logconcave distributions, which satisfy this definition with $\alpha = 1$ (see e.g. [SW14, §5.1] or [Led01, Thm 2.15]). This class includes the standard Gaussian distribution, the uniform distribution on the unit d-dimensional sphere and more generally, uniform distribution on any convex body with smooth boundaries (e.g., Gaussian smoothening of arbitrary convex bodies).

Throughout this section, let $\mathcal{C}$ be the class of functions of p halfspaces over $\mathbb{R}^d$, i.e. functions $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) = g(\mathrm{sign}(\langle w^1, x\rangle + \theta_1), \ldots, \mathrm{sign}(\langle w^p, x\rangle + \theta_p)) \tag{6.5.1}$$

for some $w^1, \ldots, w^p \in \mathbb{R}^d$ (where we use superscripts to avoid confusion with coordinate notation), $\theta_1, \ldots, \theta_p \in \mathbb{R}$, and $g : \{-1, 1\}^p \to \{-1, 1\}$. We focus on the setting where p is a constant. Also let D_X be some fixed distribution that is anticoncentrated and α-strictly subexponential. We will prove the following theorem, stated earlier as Theorem 6.1.4.

Theorem 6.5.2. *Let $\mathcal{C}$ be the class of functions of p halfspaces over $\mathbb{R}^d$, as above. Assume that $p = O(1)$. Let D_X be a distribution that is anticoncentrated and α-strictly subexponential. Then $\mathcal{C}$ can be testably learned w.r.t. D_X up to excess error ϵ using $d^{\widetilde{O}(\epsilon^{-(1+\alpha)/\alpha})}$ sample and time complexity.*

In particular whenever $\alpha = 1$, as for strongly logconcave distributions including $\mathcal{N}(0, I_d)$, we obtain a $d^{\widetilde{O}(1/\epsilon^2)}$-time algorithm.

We now describe our proof plan. To use Theorem 6.4.5, we must show that approximately matching the low-degree moments of D_X fools functions of halfspaces.

Work due to [KM13] introduced an argument for this problem based on general techniques from the classical theory of moments and the method of metric distances in probability [KR96, RKSF13]. Their broad proof approach was to use [KR96, Thm 2] to show that closeness in moments of two distributions implies closeness in the λ-distance (Definition 6.5.3), and then relate this to the CDF distance, which directly relates to fooling halfspaces. For our purposes, a direct application of [KR96, Thm 2] does not suffice. Instead, we directly analyze the λ-distance under the assumption that the moments of D_X grow in a strictly subexponential fashion. We begin with the technical lemmas we need, and then prove Theorem 6.5.2 in the final subsection.

6.5.1 Moment closeness implies distribution closeness

Definition 6.5.3 (λ-distance, see e.g. [Zol84], [RKSF13, Chap 10])**.** For a distribution P on $\mathbb{R}^p$, let $\phi_P : \mathbb{R}^p \to \mathbb{C}$ given by $\phi_P(t) = \mathbb{E}_{z \sim P}[e^{i\langle t, x\rangle}]$ be its characteristic function. For two distributions P, P' on $\mathbb{R}^p$, define the λ-distance between them as follows:

$$d_\lambda(P, P') = \min_{T>0} \max\left\{ \max_{\|t\| \le T} |\phi_P(t) - \phi_{P'}(t)|, \frac{1}{T} \right\}.$$

We prove an approximate version of [KR96, Thm 1] (see also [RKSF13, Thm 10.3.4]), bounding the λ-distance between two distributions whose moments approximately match and grow with the degree k in a strictly subexponential fashion, i.e. as $k^{k/(1+\alpha)}$.

Lemma 6.5.4. *Let $k \in \mathbb{N}$ be even. Let P be a distribution on $\mathbb{R}^p$ such that for all $\|u\| = 1$,*

$$\mathbb{E}_{z \sim P}[|\langle u, z\rangle|^k] \le M_k := p^{k/2} C_2^k k^{k/(1+\alpha)}$$

for some constant C_2. Let P' be a distribution that approximately matches moments up to order k with P in the following strong sense: for all $j \le k$ and $\|u\| = 1$,

$$\left| \mathbb{E}_{z \sim P}[\langle u, z\rangle^j] - \mathbb{E}_{z' \sim P'}[\langle u, z'\rangle^j] \right| \le \eta_j := \frac{j!}{2k} \left(\frac{6M_k}{k!} \right)^{(j+1)/(k+1)} = \frac{j!}{2k} \left(\frac{p^{\alpha}}{C_4 k^{\alpha/(1+\alpha)}} \right)^{j+1}$$

for some constant C_4 depending only on C_2. Then

$$d_\lambda(P, P^\emptyset) \;.\; \left(\frac{p}{\bar{p}k}\right)^{\alpha/(1+\alpha)}.$$

Proof. To control $d_\lambda(P, P^\emptyset)$, we need to control $\max_{\|t\| \le T} |\phi_P(t) - \phi_{P^\emptyset}(t)|$ as a function of T. To this end, fix any direction $u \in \mathbb{R}^p$ with $\|u\| = 1$, and let $t = \tau u$ for $\tau \in [0, T]$ be a vector in that direction satisfying $\|t\| \le T$. Let $\phi_1(\tau) = \phi_P(\tau u)$ and $\phi_2(\tau) = \phi_{P^\emptyset}(\tau u)$ be the characteristic functions of P and $P^\emptyset$ along u. We may Taylor expand $\phi_1 - \phi_2$ up to degree k as follows:

$$\phi_1(\tau) - \phi_2(\tau) = \sum_{0 \le j < k} \frac{\phi_1^{(j)}(0) - \phi_2^{(j)}(0)}{j!}\tau^j + \frac{\phi_1^{(k)}(\tau^\emptyset) - \phi_2^{(k)}(\tau^\emptyset)}{k!}\tau^k \tag{6.5.2}$$

for some $\tau^\emptyset \in [0, \tau]$.

The crucial fact we use now is that the derivatives of the characteristic function encode its moments. Indeed, for any τ,

$$\phi_1(\tau) = \mathbb{E}_{z \sim D}[e^{i\tau\langle z, u\rangle}] \Rightarrow \phi_1^{(j)}(\tau) = \mathbb{E}_{z \sim P}[i^j \langle z, u\rangle^j e^{i\tau\langle z, u\rangle}],$$

so that in particular $|\phi_1^{(j)}(0)| = |\mathbb{E}_{z \sim P}[\langle z, u\rangle^j]|$ for all j (and similarly for ϕ_2). This means $|\phi_1^{(0)}(0) - \phi_2^{(0)}(0)| = 0$, and for each $1 \le j < k$, by our assumption that $P^\emptyset$ approximately moment matches P, we have

$$|\phi_1^{(j)}(0) - \phi_2^{(j)}(0)| \le \eta_j. \tag{6.5.3}$$

At degree k, we have

$$|\phi_1^{(k)}(\tau^\emptyset)| = |\mathbb{E}_{z \sim P}[i^k \langle z, u\rangle^k e^{i\tau^\emptyset\langle z, u\rangle}]| \le \mathbb{E}_{z \sim P}[|\langle z, u\rangle|^k] \le M_k. \tag{6.5.4}$$

And since $\mathbb{E}_{z^\emptyset \sim P^\emptyset}[|\langle z^\emptyset, u\rangle^k|] \le \mathbb{E}_{z \sim P}[|\langle z, u\rangle^k|] + \eta_k$, we similarly have

$$|\phi_2^{(k)}(\tau^\emptyset)| \le M_k + \eta_k \le 2M_k \tag{6.5.5}$$

Substituting Eqs. (6.5.3) to (6.5.5) into Eq. (6.5.2), we obtain

$$|\phi_1(\tau) - \phi_2(\tau)| < \sum_{1 \le j < k} \frac{\eta_j}{j!}\tau^j + \frac{3M_k}{k!}\tau^k =: F(\tau), \tag{6.5.6}$$

where we have denoted the expression on the RHS by $F(\tau)$ for convenience. Since $F(\tau)$ is clearly increasing in τ and independent of u, we have $\max_{\|t\|\le T} |\phi_P(t) - \phi_{P^0}(t)| < F(T)$. This means that

$$d_\lambda(P, P^0) \le \max\left\{\max_{\|t\|\le T} |\phi_P(t) - \phi_{P^0}(t)|, \frac{1}{T}\right\} \le \max\left\{F(T), \frac{1}{T}\right\},$$

and our job now is to pick $T > 0$ that minimizes the RHS.

This is equivalent to picking the largest T such that $F(T) \ge \frac{1}{T}$, i.e.

$$TF(T) = \sum_{1 \le j<k} \frac{\eta_j}{j!} T^{j+1} + \frac{3M_k}{k!} T^{k+1} \le 1.$$

Let us divide this further into two sufficient conditions:

$$\sum_{1 \le j<k} \frac{\eta_j}{j!} T^{j+1} \le \frac{1}{2} \quad \text{and} \quad \frac{3M_k}{k!} T^{k+1} = \frac{1}{2}.$$

The second condition is equivalent to

$$T = \left(\frac{k!}{6M_k}\right)^{1/(k+1)} = \left(\frac{k!}{6p^{k/2} C_2^k k^{k/(1+\alpha)}}\right)^{1/(k+1)} \ge \frac{k^{\alpha/(1+\alpha)}}{p\bar{p}},$$

by Stirling's approximation. As for the first, we have picked η_j exactly such that when we plug in this value of T, for each $1 \le j < k$ we have

$$\eta_j = \frac{j!}{2k} T^{-(j+1)} = \frac{j!}{2k} \left(\frac{6M_k}{k!}\right)^{(j+1)/(k+1)} \Rightarrow \frac{\eta_j}{j!} T^{j+1} = \frac{1}{2k}.$$

Summing over $1 \le j < k$ verifies the first condition. Thus for this T, we have

$$d_\lambda(P, P^0) \le \max\left\{F(T), \frac{1}{T}\right\} \le \frac{1}{T} \le p\bar{p}k^{-\alpha/(1+\alpha)},$$

proving the lemma. □

We offer some remarks to guide the reader through these calculations. For our application, P will be the distribution of $\langle x, v\rangle$ for $x \sim D_X$ and $\|v\| = p\bar{p}$. The key idea is simply to use a Taylor approximation of the λ-distance to reduce the issue to one of moment closeness. The final calculation amounts to solving for T satisfying

T^{k+1} h $\frac{k!}{M_k}$, where the denominator is the k^{th} moment of P. We then set each η_j small enough to make the lower degree terms minor. Note that the j^{th} moment of P scales as $M_j = p^{j/2}C_2^j j^{j/(1+\alpha)}$, and we have $M_k^{j/k} = p^{j/2}C_2^j k^{j/(1+\alpha)}$. Thus loosely speaking, the slack η_j may be viewed in relative terms as follows:

$$\frac{\eta_j}{M_j} = \frac{1}{2k}\frac{j!}{M_j}\left(\frac{6M_k}{k!}\right)^{(j+1)/(k+1)} \tag{6.5.7}$$

$$\frac{1}{2k}\frac{j!}{M_j}\left(\frac{6M_k}{k!}\right)^{j/k} \tag{6.5.8}$$

$$\frac{1}{2k}\frac{j^j}{p^{j/2}C_2^j j^{j/(1+\alpha)}}\frac{p^{j/2}C_2^j k^{j/(1+\alpha)}}{k^j} \tag{6.5.9}$$

$$\frac{1}{2k}\left(\frac{j}{k}\right)^{j\alpha/(1+\alpha)}. \tag{6.5.10}$$

This relative slack factor is only about $1/\operatorname{poly}(k)$ for $j = O(1)$ but for $j = \Theta(k)$ it becomes $\exp(\ \Theta(k))$, which seems unavoidable with our method. Finally, note also that if P's moments scaled only as a subexponential instead of a strictly subexponential distribution, i.e. if the k^{th} moment of P scaled with k as k^k, then the key calculation for T becomes vacuous. More involved techniques (see e.g. [RKSF13, Thm 10.3.1]) still have have something to say in this situation when certain stricter moment conditions hold, but the direct Taylor expansion approach fails.

The following lemma is a convenient distillation of the rest of the argument from [KM13], where the λ-distance is related to the Levy distance (using [Gab81]), which in turn is related to the CDF distance (using anticoncentration), and which leads finally to the desired conclusion.

Lemma 6.5.5 (Implicit in [KM13, §3.3]). *Let $f : \mathsf{R}^d$! R be a function of p halfspaces as above, and also let distributions D, D^0 on R^d and P, P^0 on R^p be as above. Assume that for any continuous interval T R, each coordinate z_j of z P satis es $\mathsf{P}[z_j$ 2 $T]$ $\Theta(jTj)$. Suppose that $d_\lambda(P, P^0)$ δ. Let $N(\delta)$ be such that $\mathsf{P}_{z\ P}[kzk_1 > N(\delta)]$ δ and $\mathsf{P}_{z^0\ P^0}[kz^0k_1 > N(\delta)]$ δ. Then*

$$j\mathop{\mathsf{E}}_{D}[f]\quad \mathop{\mathsf{E}}_{D^0}[f]j\quad O\big(2^p\delta\big(\log N(\delta) + 2\log(1/\delta)\big)^p.$$

6.5.2 Approximate low-degree moment matching fools functions of halfspaces

We now prove our main structural result, which is that any distribution that approximately matches the low-degree moments of D_X fools functions of halfspaces.

Theorem 6.5.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be of the form in Eq. (6.5.1). Let D_X be a distribution that is anticoncentrated and α-strictly subexponential. For any $k \in \mathbb{N}$, let $\Delta \in \mathbb{R}_+^{|I|\le(k,d)|}$ be such that for each $I \in I(k,d)$ with $|I| = j$,*

$$\Delta_I = \frac{p}{2k}\frac{\bar{p}\, j!}{d^j}\left(\frac{1}{C_4 k^{\alpha/(1+\alpha)}}\right)^{j+1} \tag{6.5.11}$$

for some constant $C_4 > 0$. Then for any distribution D' whose moments up to order k match those of D_X up to Δ, we have

$$|\mathop{\mathbb{E}}_{D}[f] - \mathop{\mathbb{E}}_{D'}[f]| \le k^{-\alpha/(1+\alpha)}\sqrt{p\bar{p}}\left(C\log\left(\sqrt{p\bar{p}}k^{\alpha/(1+\alpha)}\right)\right)^{2p}$$

for some constant C. In particular, for $p = O(1)$, we have $|\mathbb{E}_D[f] - \mathbb{E}_{D'}[f]| \le \widetilde{O}(k^{-\alpha/(1+\alpha)})$.

Proof. Let $D = D_X$. Assume without loss of generality that $w^1, \dots, w^p$ are unit vectors, and let $W \in \mathbb{R}^{p \times d}$ be the matrix with the w^i as its rows. Let P be the distribution (on $\mathbb{R}^p$) of Wx for $x \sim D$, and define P' similarly. We would like to apply Lemma 6.5.4 to P and P'. To do so, we must first verify moment closeness. Let $u \in \mathbb{R}^p$ be a unit vector, and let $v = W^T u \in \mathbb{R}^d$. For any multi-index $I \in \mathbb{N}^d$, let v_I denote $\prod_{j\in[d]} |v_j|^{I_j}$. Then for any j,

$$\mathop{\mathbb{E}}_{z\sim P}[\langle z, u\rangle^j] = \mathop{\mathbb{E}}_{x\sim D}[\langle x, v\rangle^j] \tag{6.5.12}$$

$$= \mathop{\mathbb{E}}_{x\sim D}\Big[\sum_{|I|=j} x_I v_I\Big] \tag{6.5.13}$$

$$= \sum_{|I|=j} v_I \mathop{\mathbb{E}}_{D}[x_I]. \tag{6.5.14}$$

We place a crude upper bound on each $|v_I|$ as follows. Since W has Frobenius norm $\|W\|_F = \sqrt{p}$, we have

$$\|v\|_1 \quad \|v\| \quad \|u\|\|W\|_F \quad \sqrt{p},$$

so that in particular for each I with $|I| = j$, $|v_I| = \prod_{j \in [d]} |v_j|^{I_j} \quad \|v\|_1^{|I|} \quad p^{j/2}$. Thus

$$\begin{aligned}
|\mathop{\mathbb{E}}_{z \sim P}[\langle z, u\rangle^j] \quad \mathop{\mathbb{E}}_{z^0 \sim P^0}[\langle z^0, u\rangle^j]| &= |\mathop{\mathbb{E}}_{x \sim D}[\langle x, v\rangle^j] \quad \mathop{\mathbb{E}}_{x^0 \sim D^0}[\langle x^0, v\rangle^j]| && (6.5.15)\\
&\sum_{|I|=j} |v_I| \, |\mathop{\mathbb{E}}_{D}[x_I] \quad \mathop{\mathbb{E}}_{D^0}[x_I]| && (6.5.16)\\
&d^j p^{j/2} \sup_{|I|=j} \Delta_I && (6.5.17)\\
&\eta_j, && (6.5.18)
\end{aligned}$$

where η_j is as defined in Lemma 6.5.4, and the final inequality follows since we have picked Δ in the theorem statement precisely such that $\sup_{|I|=j} \Delta_I = d^j p^{j/2} \eta_j$. Also observe that

$$\mathop{\mathbb{E}}_{z \sim P}[\langle u, z\rangle^k] = \mathop{\mathbb{E}}_{x \sim D}[\langle x, v\rangle^k] = \|v\|^k C_2^k k^{k/(1+\alpha)} \quad p^{k/2} C_2^k k^{k/(1+\alpha)}$$

Now we apply Lemma 6.5.4 to conclude that $d_\lambda(P, P^0) \quad \sqrt{p}\bar{p}k^{\alpha/(1+\alpha)}$.

To finish the proof, we appeal to Lemma 6.5.5. For this we must first verify anticoncentration of P and estimate $N(\delta)$ as defined in that lemma. Observe first that for any $i \in [d]$, the i^{th} coordinate of $z \sim P$ (resp. $z^0 \sim P^0$) is precisely $\langle w^i, x\rangle$ for $x \sim D$ (resp. $\langle w^i, x^0\rangle$ for $x^0 \sim D^0$). Anticoncentration of each coordinate of z follows immediately from Definition 6.5.1(c). To estimate $N(\delta)$, we will use a simple Chebyshev-style bound. For any coordinate $i \in [d]$ and any even degree $j \leq k$, we have

$$\mathop{\mathbb{P}}_{D}[|\langle w^i, x\rangle| > t] \quad \frac{\mathop{\mathbb{E}}_{D}[\langle w^i, x\rangle^j]}{t^j} \quad \frac{C_2^j j^{j/(1+\alpha)}}{t^j}.$$

And since D^0 approximately matches moments with D, by a similar calculation as earlier (now with $\|w^i\|_1 \quad 1$ in place of $\|v\|_1 \quad \sqrt{p}$),

$$\mathop{\mathbb{E}}_{D^0}[\langle w^i, x^0\rangle^j] \quad \mathop{\mathbb{E}}_{D}[\langle w^i, x\rangle^j] + \sum_{|I|=j} |w_I| \Delta_I \quad \mathop{\mathbb{E}}_{D}[\langle w^i, x\rangle^j] + \eta_j/p^{j/2} \quad 2\mathop{\mathbb{E}}_{D}[\langle w^i, x\rangle^j],$$

and so

$$\mathop{\mathsf{P}}_{D^0}[|\langle w^i, x^0\rangle| > t] \quad \frac{\mathsf{E}_{D^0}[\langle w^i, x^0\rangle^j]}{t^j} \quad \frac{2C_2^j j^{j/(1+\alpha)}}{t^j} = 2\left(\frac{C_2 j^{1/(1+\alpha)}}{t}\right)^j.$$

We need t such that the RHS is at most δ/p. For this it suffices to set $j = 2\log(p/\delta)$ and $t = C_2 j^{1/(1+\alpha)}$ for this. By a union bound over the p coordinates of z P (similarly z^0 P^0), we see that we may take $N(\delta) = t = O((\log(p/\delta))^{1/(1+\alpha)})$.

We are now ready to apply Lemma 6.5.5 with this $N(\delta)$ and δ $\mathsf{p}\overline{p}k^{\alpha/(1+\alpha)}$. Substituting these expressions in, we get that

$$\begin{aligned}
|\mathop{\mathsf{E}}_{D}[f] \quad \mathop{\mathsf{E}}_{D^0}[f]| \quad & O\big(2^p\delta\big(\log N(\delta) + 2\log(1/\delta)\big)^p\big) && (6.5.19)\\
& 2^p\delta\Big(C^0\log(\frac{1}{\delta}\log\frac{p}{\delta})\Big)^p && \text{(for some constant } C^0 > 0)\\
& \delta\Big(C\log(\frac{p}{\delta})\Big)^{2p} && \text{(for some constant } C > 0)\\
& k^{\alpha/(1+\alpha)}\mathsf{p}\overline{p}\big(C\log(\mathsf{p}\overline{p}k^{\alpha/(1+\alpha)})\big)^{2p}, && (6.5.20)
\end{aligned}$$

as claimed. □

We pause to note an interesting corollary of this theorem, stated informally earlier as Corollary 6.1.5, which states that any D^0 that approximately matches low-degree moments with D_X must be anticoncentrated.

Corollary 6.5.7. *Let $\epsilon > 0$, $k = \widetilde{O}(\epsilon^{-(1+\alpha)/\alpha})$, and Δ be as in Theorem 6.5.6, with $p = 2$. Let D^0 be any distribution whose moments up to order k match those of D_X up to Δ. Then for any $\|u\| = 1$ and any continuous interval T R, $\mathsf{P}_{x\ D^0}[\langle x, u\rangle 2\ T]$ $\mathsf{P}_{x\ D_X}[\langle x, u\rangle 2\ T] + \epsilon$ $\Theta(|T|) + \epsilon$.*

Proof. Write $T = [\theta, \theta^0]$ for some $\theta < \theta^0$ 2 R, and consider the function $f(x) = \text{sign}(\langle x, u\rangle - \theta) \wedge \text{sign}(\theta^0 - \langle x, u\rangle)$ (where $b_1 \wedge b_2 = 1$ iff $b_1 = b_2 = 1$). Clearly $\langle x, u\rangle$ 2 T iff $f(x) = 1$. But f is an intersection of two halfspaces, and we know by Theorem 6.5.6 that $|\mathsf{E}_{D^0}[f] - \mathsf{E}_{D_X}[f]| \epsilon$. Since $\mathsf{E}_{D_X}[f]$ $C_3|T|$ by Definition 6.5.1(c), the statement follows. □

In fact, the same reasoning tells us that for any collection of $p = O(1)$ intervals T and any $\|u\| = 1$, $\mathbb{P}_{D^0}[\langle x, u\rangle \in T] = \mathbb{P}_{D_X}[\langle x, u\rangle \in T] \pm \epsilon$.

6.5.3 Proof of Theorem 6.5.2

The final ingredient for the proof of Theorem 6.5.2 is the following lemma, which gives a bound on the sample complexity required for the empirical moments of D_X to concentrate about their true moments.

Lemma 6.5.8. *Let the degree parameter be k, and the slack parameter $\Delta \in \mathbb{R}_+^{I(k,d)}$ be as in Theorem 6.5.6. Assume $p = O(1)$. Then drawing a sample of size $m = d^{\widetilde{O}(k)}$ from D_X is sufficient to ensure that with high probability, the empirical moments of order at most k match those of D_X up to slack Δ.*

Proof. Let $D = D_X$, and let $\widehat{D}_m$ denote the empirical distribution on a sample S of size m drawn from D. We would like to ensure that for every $I \in I(k,d)$, $|\mathbb{E}_{\widehat{D}_m}[x_I] - \mathbb{E}_D[x_I]| \le \Delta_I$. It suffices to consider the case when x_I has the weakest concentration, and this is clearly when $|I| = k$ and in fact when $I = (k, 0, \ldots, 0)$ (without loss of generality), so that $x_I = x_1^k$.

Let Z denote the random variable $\mathbb{E}_{\widehat{D}_m}[x_1^k] - \mathbb{E}_D[x_1^k]$. For our purposes it is sufficient to use a crude Chebyshev bound, although higher moment analogs will give a slightly better bound. We have $\mathrm{Var}[Z] = \frac{1}{m}\mathrm{Var}[x_1^k] \le \frac{1}{m} C_2^{2k}(2k)^{2k/(1+\alpha)}$. Thus

$$\mathbb{P}[Z > \Delta_I] \le \frac{\mathrm{Var}[Z]}{\Delta_I^2} \tag{6.5.21}$$

$$\frac{1}{m}\frac{C_2^{2k}(2k)^{2k/(1+\alpha)}}{\Delta_I^2} \tag{6.5.22}$$

$$\frac{1}{m} C_2^{2k}(2k)^{2k/(1+\alpha)} \left(\frac{2k}{p}\frac{d^k}{k!}\right)^2 (C_4 k^{\alpha/(1+\alpha)})^{2(k+1)} \tag{6.5.23}$$

$$\frac{1}{m}(dk)^{O(k)}, \tag{6.5.24}$$

after plugging in the value of Δ_I when $|I| = k$ from Eq. (6.5.11) and some manipulation. This is at most δ if $m \ge (dk)^{O(k)}/\delta$.

For moment closeness to hold simultaneously for all $I \ 2 \ I(k,d)$ with high probability, we set $\delta = \Theta(1/jI(k,d)j) = d^{(k)}$ and apply a union bound. For this δ, m may be simplified to $d^{\widetilde{O}(k)}$, as desired. □

We are now ready to prove Theorem 6.5.2 using our general algorithm, Theorem 6.4.5.

Proof of Theorem 6.5.2. We need to pick m, k and Δ suitably as functions of ϵ and d, and verify that the conditions in Theorem 6.4.5 hold. Let Δ be as defined in Theorem 6.5.6. By Lemma 6.5.8, it suffices to take $m = d^{\widetilde{O}(k)}$ to ensure that with high probability, the empirical distribution $\widehat{D}_m$ matches moments of order at most k with $N(0, I_d)$ up to Δ. This verifies condition (a) of Theorem 6.4.5. Now for any $f \ 2 \ C$, we can combine Theorem 6.5.6 with Theorem 6.3.2 to obtain degree-k sandwiching polynomials satisfying condition (b) of Theorem 6.4.5, with $\epsilon = \widetilde{O}(k^{-\alpha/(1+\alpha)})$, or equivalently $k = \widetilde{O}(\epsilon^{-(1+\alpha)/\alpha})$. Applying Theorem 6.4.5 completes the proof. □

6.6 Sample complexity of testable learning

In this section we show that the sample complexity of testably learning a class is characterized by its Rademacher complexity. Throughout this section, let D_X be the target marginal, and let C be the concept class (mapping X to $f \ 1g$) that we wish to testably learn w.r.t. D_X. We remind the reader that we use the term "with high probability" to mean "with probability at least 0.99" for simplicity.

6.6.1 Upper bound

We begin with the upper bound, which is essentially just the observation that the empirical Rademacher complexity provides a generalization guarantee that can be estimated to high accuracy from the sample itself. Note that this is an information-theoretic upper bound.

Theorem 6.6.1. *Let $\epsilon > 0$, and let m^0 be such that $R_{m^0}(C, D_X)$ $\frac{\epsilon}{5}$. Then C can be testably learned w.r.t. D_X up to excess error ϵ with sample complexity $m^0 + O(1/\epsilon^2)$.*

Proof. Let $m = m^0 + O(1/\epsilon^2)$. Let $S = f(x_i, y_i)g_{i2[m]}$ D^m be a sample of m labeled points drawn from D, and let S_X denote $fx_ig_{i2[m]}$. Our tester T accepts iff $\widehat{R}_m(C, S_X)$ $\frac{\epsilon}{4}$. Whenever the tester accepts, the learner A simply performs ERM over S w.r.t. C.

To see why completeness is satisfied, suppose that the true marginal is in fact D_X. Since $R_m(C, D_X)$ $R_{m^0}(C, D_X)$ $\frac{\epsilon}{5}$, by Eq. (6.2.3) we can ensure that with high probability over S_X, $\widehat{R}_m(C, S_X)$ $\frac{\epsilon}{4}$, and so T will accept.

Soundness holds by a standard argument showing generalization using uniform convergence. Formally, suppose that the tester accepts S_X, i.e. $\widehat{R}_m(C, S_X)$ $\frac{\epsilon}{4}$. Consider an ERM hypothesis

$$\widehat{f}_m = \arg\min_{f2C} \frac{1}{m} \sum_{i2[m]} \ell(f(x_i), y_i)$$

as well as an optimal hypothesis

$$f \ = \arg\min_{f2C} \ \mathsf{E}_{(x,y)\ D}[\ell(f(x), y)].$$

Then with high probability over S we have

$$\begin{aligned}
L(\widehat{f}_m) \quad & \widehat{L}_m(\widehat{f}_m) + \widehat{R}_m(C) + O\left(\sqrt{\frac{1}{m}}\right) && \text{(by Eq. (6.2.5))} \\
& \widehat{L}_m(f\ ) + \widehat{R}_m(C) + O\left(\sqrt{\frac{1}{m}}\right) && (\widehat{f}_m \text{ is an ERM hypothesis}) \\
& L(f\ ) + 2\widehat{R}_m(C) + O\left(\sqrt{\frac{1}{m}}\right) && \text{(by Eq. (6.2.5) again)} \\
& L(f\ ) + \frac{\epsilon}{2} + \frac{\epsilon}{2} = L(f\ ) + \epsilon, && (6.6.1)
\end{aligned}$$

by our choice of m. This proves the theorem. □

6.6.2 Lower bound

Now we state the lower bound, which matches the upper bound up to a quadratic factor. Our lower bound can be viewed as a generalization of the argument of [RV22], who proved lower bounds for testable learning for the specific cases of convex sets and monotone functions. We obtain the full range of lower-bounds for any value of ϵ purely in terms of Rademacher complexity. The idea here is that no tester with bounded sample complexity m can distinguish between a distribution D_X and the uniform distribution on a sufficiently large sample of size M m drawn from D_X. If the Rademacher complexity w.r.t. D_X at sample size M is somewhat large, then the large sample can be labeled randomly and still admit nontrivial optimal error, but of course the learner cannot do well on unseen data.

Theorem 6.6.2. *Let $\epsilon > 0$, and let M be such that $R_M(C, D_X)$ 5ϵ. Assume that M $\Theta(1/\epsilon^2)$ is su ciently large. Then testably learning C up to excess error ϵ requires sample complexity at least $\Omega(\sqrt{M})$.*

Proof. Suppose we had a tester-learner (T, A) requiring sample complexity only m where m $\frac{\sqrt{M}}{100}$. We will show how to "fool" (T, A) into failing its guarantee by constructing a (random) labeled distribution D such that (with high probability over the randomness of our construction),

(a) $\mathsf{opt}(D, C)$ $\frac{1}{2}$ 2ϵ;

(b) with high probability, T will accept a sample of size m drawn from D; and yet

(c) with high probability, A's output will have error greater than $\frac{1}{2}$ ϵ on D.

For such a D, it is clear that the tester-learner pair (T, A) fails its guarantee in that with high probability, despite T accepting, A cannot produce a hypothesis with error at most $\mathsf{opt}(D, C) + \epsilon$.

We construct D as follows. Draw a sample of M randomly labeled points $S = f(x_i, y_i)g_{i2[M]}$ $(D_X$ Unif$f$ $1g)$ ^{M}, and let S_X denote $fx_ig_{i2[m]}$. Define D to be the uniform distribution over S. We now show that with high probability over the draw of S (including its random labeling), the distribution D satisfies the required properties.

Denote the size-m sample given to (T, A) by S^0 D ^{m}, and let S^0_X denote its marginal. That is, S^0 is an iid subsample of S. Notice that unless we sample the same element of S twice, S^0_X is distributed exactly as a sample of m points drawn directly from D_X. In other words, the statistical distance between S^0_X formed in this way vs by drawing m points directly from D_X is at most the collision probability, which is m^2/M 10 ^{4}. Thus any tester satisfying completeness must accept S^0 with high probability. This gives us property (b).

Let us see why property (a) holds with high probability over S. The idea is that because $R_M(C)$ $\Omega(\epsilon)$, we expect that there exists a classifier in C that achieves error at most $\frac{1}{2}$ $\Omega(\epsilon)$ on the randomly labeled sample S. Indeed, observe that the random labels are exactly equivalent to Rademacher random variables. Assuming that M is sufficiently large and applying Eqs. (6.2.3) and (6.2.4) successively, we obtain that with high probability over the sample S (together with the realization of the random labels),

$$\left| R_M(C) \quad \sup_{f2C} \frac{1}{M} \sum_{i=1}^{M} y_i f(x_i) \right| \quad \epsilon.$$

In particular, since $R_M(C)$ 5ϵ, there exists f 2 C such that $\frac{1}{M}\sum_{i=1}^{M} y_i f\ (x_i)$ 4ϵ, or equivalently

$$\frac{1}{M}\sum_{i=1}^{M} \mathbb{1}[f\ (x_i)\ 6= y_i] = \frac{1}{M}\sum_{i=1}^{M} \frac{1 \quad y_i f\ (x_i)}{2} = \frac{1}{2} \quad \frac{1}{2M}\sum_{i=1}^{M} y_i f\ (x_i) \quad \frac{1}{2} \quad 2\epsilon.$$

In other words, $\mathrm{opt}(D, C)$ $\frac{1}{2}$ 2ϵ.

For property (c), the idea is that the learner, having only seen a minuscule fraction of the randomly labeled D, cannot possibly output a hypothesis with error

substantially better than $\frac{1}{2}$ on all of $\mathcal{D}$. Formally, observe that any classifier h that A outputs is stochastically independent of $S \cap S'$. This means that in expectation over S and the randomness of A,

$$\mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) = y] \geq \frac{m}{M} + \frac{1}{2}\left(\frac{M-m}{M}\right) = \frac{1}{2} + \frac{m}{2M} \geq \frac{1}{2} + \frac{\epsilon}{2},$$

since the fact that $M \geq \Theta(1/\epsilon^2)$ and $m \geq \sqrt{M}/100$ mean that $\frac{m}{M} \geq \frac{1}{100\sqrt{M}} > \epsilon$. It is clear that for sufficiently large $M \geq \Theta(1/\epsilon^2)$, we will actually have $\mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) = y] > \frac{1}{2} + \epsilon$ with high probability over S and A. (Such an argument is also formalized as [RV22, Lemma 25].)

□

Note that this theorem becomes stronger if ϵ is taken to be a constant. In particular, if we assume $R_M(\mathcal{C}, D_X) \geq 0.99$, then the same argument would actually yield a "fooling distribution" $\mathcal{D}$ such that

(a) $\mathsf{opt}(\mathcal{D}, \mathcal{C}) \geq 0.01$;

(b) with high probability, T will accept a sample of size m drawn from $\mathcal{D}$;

(c) with high probability, A's output will have error greater than 0.49 on $\mathcal{D}$.

This would rule out any tester-learner capable of testably learning up to error sufficient to distinguish the case where $\mathsf{opt}(\mathcal{D}, \mathcal{C}) = 0.01$ from $\mathsf{opt}(\mathcal{D}, \mathcal{C}) > 0.49$ (e.g., one with final error guarantee $10 \cdot \mathsf{opt}(\mathcal{D}, \mathcal{C}) + 0.1$).

We also give the following stronger version of this lower bound, stated in terms of the behavior of the empirical Rademacher complexity (which is a random variable depending on the sample). This is a very strong lower bound that holds whenever $\widehat{R}_M(\mathcal{C}, S_X) = 1$ with high probability, because it yields a fooling distribution $\mathcal{D}$ that is in fact perfectly realizable. As we will see in Section 6.6.2.1, this turns out to apply to convex sets and monotone functions. In a sense, this version is not really about

Rademacher complexity but rather the stronger notion of shattering (except with high probability over a sample, like a distribution-specific version of the VC dimension, albeit stronger than VC entropy). Recall that C is said to shatter an unlabeled set S_X if every possible labeling of S_X can be achieved by some $f \in C$, or equivalently $\widehat{R}_M(C, S_X) = 1$.

Theorem 6.6.3. *Let M be such that with high probability over a size-M sample $S_X \sim D_X{}^M$, $\widehat{R}_M(C, S_X) = 1$, i.e. C shatters S_X. Consider any tester-learner pair (T, A) for testably learning C up to error su cient to distinguish the case where $\mathrm{opt}(D, C) = 0$ from $\mathrm{opt}(D, C) > 0.49$. Then (T, A) requires sample complexity at least $\Omega(\sqrt{M})$.*

In particular, this rules out any tester-learner with nal error guarantee $\psi(\mathrm{opt}(D, C)) + 0.49$ for any increasing function $\psi : [0, 1] \to \mathbb{R}$ satisfying $\psi(0) = 0$.

Proof. The proof is a simpler version of the earlier one. Again, suppose we had a tester-learner (T, A) requiring sample complexity only $m \leq \frac{\sqrt{M}}{100}$. We construct a labeled distribution D such that

(a) $\mathrm{opt}(D, C) = 0$;

(b) with high probability, T will accept a sample of size m drawn from D; and yet

(c) with high probability, A's output will have error greater than 0.49 on D.

The distribution D is constructed in exactly the same way: draw a sample of M randomly labeled points $S = \{(x_i, y_i)\}_{i \in [M]}$ $(D_X = \mathrm{Unif}\{1\})^M$, and define D to be the uniform distribution over S. Let $S_X = \{x_i\}_{i \in [M]}$. Let $S' \sim D^m$ denote the sample given to (T, A), and as before, let us condition on its marginal S'_X containing no duplicates (which occurs with high probability).

Property (a) follows immediately from our assumption that with high probability, C shatters S_X. (Note that this also implies that S_X contains no duplicates.)

Properties (b) and (c) follow by almost exactly the same arguments as before (for the latter, we now use the fact that $m/M \geq 1/100$ instead of $m/M \geq \epsilon$). $\square$

6.6.2.1 Applications

The lower bounds of [RV22] may be viewed as applications of Theorem 6.6.3. The first application is the class of convex sets w.r.t. $\mathsf{N}(0, I_d)$, and the second is the class of monotone Boolean functions w.r.t. $\mathrm{Unif}\{-1,1\}^d$.

Theorem 6.6.4 (Implicit in [RV22, Theorem 22]). *Let $X = \mathbb{R}^d$, $D_X = \mathsf{N}(0, I_d)$, and C be the class of ($\{-1,1\}$-valued indicator functions of) convex sets in $\mathbb{R}^d$. Let $M = 2^{Cd}$ for some small constant $C > 0$. Then with probability $1 - \exp(-\Omega(d))$ over the draw of a size-M sample $S \sim D_X^M$, C can shatter S.*

Theorem 6.6.5 (Implicit in [RV22, Theorem 23]). *Let $X = \{-1,1\}^d$, $D_X = \mathrm{Unif}\{-1,1\}^d$, and C be the class of monotone Boolean functions. Let $M = 2^{Cd}$ for some small constant $C > 0$. Then with probability $1 - \exp(-\Omega(d))$ over the draw of a size-M sample $S \sim D_X^M$, C can shatter S.*

In fact, Rubinfeld and Vasilyan are able to state their lower bounds in a slightly stronger way because of the specific parameters M, m that these examples above allow. Specifically, for both convex sets and monotone functions, we may take $M = 2^{\Theta(d)}, m = M^{0.01} = 2^{\Theta(d)}$, and the same argument as in Theorem 6.6.3 can be analyzed more closely to yield a distribution D such that $\mathsf{opt}(D, C) = 0$ and yet the final output of any testable learner with sample complexity m must have $\exp(-\Omega(d))$ advantage over random guessing (which is stronger than merely saying the output must have error at least 0.49).

Interestingly, these examples add to what has been called "the emerging analogy between symmetric convex sets in Gaussian space and monotone Boolean functions"; see [DNS22] and references therein.

6.7 Discussion

6.7.1 Implications for the uniform convergence paradigm

As observed in [RV22], an interesting consequence of the lower bounds in Section 6.6.2.1 is that they demonstrate a strict separation between distribution-specific agnostic learning and testable learning. In the case of both convex sets over $N(0, I_d)$ and monotone functions over Unif$\{\pm 1\}^d$, Fourier-theoretic arguments are known to give agnostic learners requiring sample complexity only $2^{\widetilde{O}(\sqrt{d}/\mathrm{poly}(\epsilon))}$ to learn up to excess error ϵ [BT96, KOS08]. In particular, they require only sample complexity $2^{\widetilde{O}(\sqrt{d})}$ to learn up to excess error $\epsilon = 0.1$ (say), which is much smaller than the lower bounds of $2^{\Omega(d)}$ for testably learning these classes up to $\epsilon = 0.1$.

But we have just characterized testable learning in terms of Rademacher complexity, which we know in turn tightly characterizes uniform convergence (Theorem 6.2.8). We draw the following implications from this:

- Uniform convergence is always sufficient for distribution-specific agnostic learning but it is not necessary, as witnessed by the examples of convex sets and monotone functions.
- Uniform convergence is both necessary and sufficient for testable learning.

That is, not only is there a strict separation between distribution-specific agnostic learning and testable learning, it is the latter that is in fact characterized by uniform convergence.

Uniform convergence in distribution-free vs distribution-specific learning. In the distribution-free setting, uniform convergence is well-known to be necessary and sufficient for agnostic (as well as realizable) learning, by classic VC theory (see e.g. [SB14, Chapter 6]). Let us clarify that in the distribution-free setting the term "uniform convergence" now means a uniform bound on the generalization gap over

not just all $f \in C$ but also all distributions D_X; that is, we now care about the *worst-case distribution-free* generalization gap: $\sup_{D_X} \sup_{f \in C} |L(f) - \widehat{L}_m(f)|$. This quantity is tightly governed by the VC-dimension of C, a distribution-free, purely combinatorial property.

Meanwhile in the distribution-specific setting, the statistical complexity of agnostic learning is known to be characterized by the $L^1(D_X)$ metric entropy of the class (aka the log covering number w.r.t. the metric $\rho(f,g) = \mathbb{E}_{x \sim D_X}[|f(x) - g(x)|]$, which for $\{-1,1\}$-valued f, g is the same as $2\mathbb{P}_{x \sim D_X}[f(x) \neq g(x)]$) [BI91]. In this setting, uniform convergence (now in the distribution-specific sense of bounds on $\sup_{f \in C} |L(f) - \widehat{L}_m(f)|$ for fixed D_X) is sufficient but not necessary. Indeed, we may also view the separations given by convex sets and monotone functions as separations between the metric entropy and the Rademacher complexity of these classes.

A priori, this seems like a surprising difference between distribution-free and distribution-specific agnostic learning. However, one could argue that the more realistic distribution-specific supervised learning model is that of testable learning. Here we see that uniform convergence is again necessary and sufficient.

Relationship to modern overparametrized models. The inadequacies of the uniform convergence paradigm have been a topic of much study in modern deep learning theory (see e.g. [ZBH+21, NK19, BMR21, Bel21]). We may phrase the essential argument in the following way. Let C be a certain "rich" concept class mapping X to $\{-1,1\}$ (for concreteness). Let D be an unknown labeled distribution on $X \times \{-1,1\}$, and let $S \sim D^m$ be a sample drawn from it. Let L and $\widehat{L}_m$ denote the population and empirical 0-1 loss functionals, as before. Consider an ERM estimator $\widehat{f}$ picked based on this sample: $\widehat{f} \in \arg\min_{f \in C} \widehat{L}_m(f)$. We are interested in the generalization gap associated with $\widehat{f}$, namely the quantity $|L(\widehat{f}) - \widehat{L}_m(\widehat{f})|$. We would like to place a useful upper bound, say B, on this quantity.

The core observation is that certain classes C (such as deep neural networks)

are rich enough that they can interpolate any sample of size m; in this sense they are "overparametrized" relative to sample size m. In particular, they can fit even completely random labels. This of course means that $\widehat{L}_m(\widehat{f}) = 0$ while $L(\widehat{f}) = \frac{1}{2}$. This in turn means that the bound B must be at least $\frac{1}{2}$. Note that this occurs without changing anything about the class C, the marginal distribution D_X of D, or the training procedure (ERM). So any bound B that is purely a function of these quantities must be essentially vacuous; this includes uniform convergence bounds (e.g., $\sup_{f \in \mathsf{C}} |L(f) - \widehat{L}_m(f)| \leq R_m(\mathsf{C}, D_X)$) as well as algorithm-based bounds (e.g. those based on stability). Yet what is remarkable is that when the labels do satisfy some structure, e.g. when there exists $f \in \mathsf{C}$ achieving error $L(f) = \mathsf{opt}(\mathsf{C}, \mathsf{D}) < \frac{1}{2}$, then we observe (provably or empirically) that the generalization gap is in fact relatively small, and $\widehat{f}$ performs comparably with f. This phenomenon, sometimes referred to as "benign overfitting", occurs not only with deep neural networks but in fact also already (in a sense) with linear regression [BLLT20, HMRT22]; we shall not attempt a summary of known results here but direct the reader to e.g. [BMR21, Bel21].

What the results in this chapter point out is that a version of this phenomenon also occurs in a strong, provable sense with classical examples such as convex sets in Gaussian space or monotone functions over the Boolean hypercube. These classes C are also capable of interpolating a random sample of size $m = 2^{\Omega(d)}$; and yet there exist estimators $\widehat{f}$ that achieve error $\mathsf{opt}(\mathsf{C}, \mathsf{D}) + \epsilon$ using sample complexity only $2^{\widetilde{O}(\overline{p}\overline{d})/\mathrm{poly}(\epsilon)}$ [BT96, KOS08]. These estimators are not based on ERM and do not lie strictly in C; instead, they are low-degree (specifically, degree-$O(\overline{p}\overline{d}/\mathrm{poly}(\epsilon))$ polynomial approximators of functions in C (as in Theorem 6.2.2). Such polynomial approximators essentially constitute a small (improper) cover of the class C (w.r.t. the $L^1(D_X)$ metric $\rho(f, p) = \mathsf{E}_{D_X}[|f - p|]$). The improved sample complexity we obtain by such methods may be explained by the fact that to obtain generalization, it suffices to have uniform convergence over this cover as opposed to all of C (as in the metric entropy characterization of [BI91]).

6.7.2 Implications for sandwiching degree

Another somewhat surprising consequence of the lower bounds in Section 6.6.2.1 is that the classes of convex sets over $\mathcal{N}(0, I_d)$ and monotone functions over $\mathrm{Unif}\{-1,1\}^d$ cannot admit sandwiching polynomials of degree $o(d/\log d)$ and error even $\epsilon = \Theta(1)$ unless they have very large coefficients. This is simply because any such sandwiching polynomials, if they have reasonable coefficients and if the distribution satisfies some concentration properties, will tend to allow the moment matching algorithm (Theorem 6.4.5) to succeed. More generally, we obtain the following surprising connection between Rademacher complexity and sandwiching degree as a direct corollary of Theorems 6.4.5 and 6.6.2.

Corollary 6.7.1. *Let $\epsilon > 0$, let D_X be a distribution on X, and let C be a concept class mapping X to $\{-1,1\}$. Let M be such that $R_M(C) \le 5\epsilon$, and assume $M \ge \Theta(1/\epsilon^2)$. Consider any degree and slack parameters $k \in \mathbb{N}$, $\Delta \in \mathbb{R}_+^{|I| \le (k,d)|}$ such that each $f \in C$ admits degree-k sandwiching polynomials $p_l \le f \le p_u$ satisfying*

$$\mathop{\mathbb{E}}_{D_X}[p_u - f] + \langle \Delta, |p_u| \rangle \le \frac{\epsilon}{2}, \qquad \mathop{\mathbb{E}}_{D_X}[f - p_l] + \langle \Delta, |p_l| \rangle \le \frac{\epsilon}{2}.$$

Let m be the sample complexity of testing with high probability whether the degree-k empirical moments of D_X are within Δ of their true moments. Then we must have $m \ge \Omega(\sqrt{M})$.

Let us illustrate this in the cases where C, D_X are either convex sets over $\mathcal{N}(0, I_d)$ or monotone functions over $\mathrm{Unif}\{-1,1\}^d$. Consider any degree and slack parameters k, Δ, and let $\delta = \min_{I \ne \emptyset, |I| \le (k,d)} \Delta_I$. In both cases, one can check with high probability whether the degree-k empirical moments of D_X are within Δ of their true moments using sample complexity at most $m \le d^{O(k)} \mathrm{poly}(1/\delta)$ (for $\mathrm{Unif}\{-1,1\}^d$ this is immediate by boundedness, while for $\mathcal{N}(0, I_d)$ we appeal to Lemma 6.5.8). Now suppose that each $f \in C$ admitted degree-k $(\epsilon/4)$-sandwiching polynomials $p_l \le f \le p_u$ satisfying $\mathbb{E}_{D_X}[f - p_l], \mathbb{E}_{D_X}[p_u - f] \le \epsilon/4$ and also such that their coefficients are bounded in magnitude by $d^{O(k)}$. Then clearly we can pick Δ sufficiently small so that

$\mathbb{E}[\Delta, p_l] \mathbb{E}[\Delta, p_u]$ $\epsilon/4$ while still ensuring δ $\epsilon d^{O(k)}$. This means m as defined earlier is $d^{O(k)} \operatorname{poly}(1/\epsilon)$. Thus for this choice of Δ, both conditions (a) and (b) of Theorem 6.4.5 hold, and we obtain a testable learning algorithm with sample complexity $m = d^{O(k)} \operatorname{poly}(1/\epsilon)$. For $\epsilon = 0.1$, say, we know by Section 6.6.2.1 that the required sample complexity for this task is $2^{(d)}$. Thus we see that k must necessarily be $\Omega(d/\log d)$.

The only way for sandwiching polynomials to exist despite this obstacle is by having unusually large coefficients (on the scale of $d^{\omega(k)}$). Most reasonable approaches to constructing sandwiching polynomials will tend to ensure some boundedness of coefficients (indeed, this is true whenever such polynomials are constructed out of univariate polynomials that are bounded on a bounded domain, see e.g. [She12, Lemma 4.1]). Therefore, we regard this as good evidence in favor of a lower bound on the sandwiching degree for these classes.

Appendix A: Superpolynomial Lower Bounds for Learning One-Layer Neural Networks using Gradient Descent

A.1 Bounding the function norms under the Gaussian

Our goal in this section will be to give lower bounds on the norms of the functions in $\mathsf{C}_{\mathsf{orth}}(n,k)$, which is a technical requirement for our results to hold (see Lemma 2.4.4 and Corollary 2.4.6). Note that when learning with respect to L_2 error, such a lower bound is necessary if we wish to state SQ lower bounds, since if the target had small norm, say $\|f\|_D \leq \epsilon$, then the zero function trivially achieves L_2 error ϵ.

All inner products and norms in this section will be with respect to the standard Gaussian, $N(0,I)$. Since we will fix S throughout, for our purposes the only relevant part of the input is x_S and so we drop the subscripts and let $g = g_S, f = f_S$ and $x = x_S$, so that g and f are functions of $x \in \mathbb{R}^k$. Our approach will be as follows. In order to prove a norm lower bound on f, we will prove an anticoncentration result for g. To this end we first calculate the second moment of g in terms of the Hermite coefficients of ϕ.

Lemma A.1.1. *Under the distribution $N(0,I_n)$, let the Hermite representation of ϕ be $\phi(x) = \sum_{i=0}^{\infty} \widehat{\phi}_i \tilde{H}_i(x)$, where $\tilde{H}_i(x)$ is the i^{th} normalized probabilists' Hermite polynomial. Then*

$$\mathbb{E}\left[g(x)^2\right] = 4^k \sum_{i=0}^{\infty} \frac{\widehat{\phi}_i^2}{k^i} \sum_{\substack{i_1+\cdots+i_k=i \\ i_1,\ldots,i_k \text{ are odd}}} \binom{i}{i_1,\ldots,i_k}.$$

Proof. We use E in this proof instead of $\mathsf{E}_{x \sim N(0,I_n)}$ for simplicity. Then we have

$$\begin{aligned}
&\mathsf{E}\left[g(x)^2\right] \\
&= \mathsf{E}\left[\sum_{\alpha \in \{0,1\}^k} \chi(\alpha)\phi\left(\frac{\alpha \cdot x_S}{\sqrt{k}}\right)\right]\left[\sum_{\beta \in \{0,1\}^k} \chi(\beta)\phi\left(\frac{\beta \cdot x_S}{\sqrt{k}}\right)\right] \\
&= \sum_{\alpha,\beta \in \{0,1\}^k} \prod_{l=1}^{k} \alpha_l\beta_l \, \mathsf{E}\left[\phi\left(\frac{\alpha \cdot x_S}{\sqrt{k}}\right)\phi\left(\frac{\beta \cdot x_S}{\sqrt{k}}\right)\right] \\
&= \sum_{\alpha,\beta \in \{0,1\}^k} \prod_{l=1}^{k} \alpha_l\beta_l \, \mathsf{E}\left[\sum_{i,j \geq 0} \widehat{\phi}_i\widehat{\phi}_j \tilde{H}_i\left(\frac{\alpha \cdot x_S}{\sqrt{k}}\right)\tilde{H}_j\left(\frac{\beta \cdot x_S}{\sqrt{k}}\right)\right] \\
&= \sum_{\alpha,\beta \in \{0,1\}^k} \prod_{l=1}^{k} \alpha_l\beta_l \sum_{i,j \geq 0} \widehat{\phi}_i\widehat{\phi}_j \, \mathsf{E}\left[\tilde{H}_i\left(\frac{\alpha \cdot x_S}{\sqrt{k}}\right)\tilde{H}_j\left(\frac{\beta \cdot x_S}{\sqrt{k}}\right)\right].
\end{aligned}$$

Since $x \sim N(0, I_k)$, $\frac{\langle \alpha, x_S \rangle}{\sqrt{k}}$ and $\frac{\langle \beta, x_S \rangle}{\sqrt{k}}$ are both standard Gaussian and have correlation $\frac{\langle \alpha, \beta \rangle}{k}$, we then apply the following well-known property of the Hermite polynomials.

$$\mathsf{E}_{(a,b)^T \sim N\left(0, \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)\right)} \tilde{H}_i(a)\tilde{H}_j(b) = \delta_{i,j}\rho^i,$$

where $\delta_{i,j}$ is the Dirac delta function.

$$\begin{aligned}
\mathsf{E}\left[g(x)^2\right] &= \sum_{\alpha,\beta \in \{0,1\}^k} \prod_{l=1}^{k} \alpha_l\beta_l \sum_{i \geq 0} \widehat{\phi}_i^2 \left(\frac{\alpha \cdot \beta}{k}\right)^i \\
&= \sum_{w,\theta \in \{0,1\}^k} \prod_{l=1}^{k} w_l \sum_{i \geq 0} \widehat{\phi}_i^2 \left(\frac{\sum_{l=1}^{k} w_l}{k}\right)^i \\
&= 2^k \sum_{w \in \{0,1\}^k} \prod_{l=1}^{k} w_l \sum_{i \geq 0} \widehat{\phi}_i^2 \left(\frac{\sum_{l=1}^{k} w_l}{k}\right)^i,
\end{aligned}$$

where $w_i = \alpha_i\beta_i$ and $\theta_i = w_i\alpha_i$. Note that Assumption 2.3.3 implies that $\sum_{i=0}^{1} \widehat{\phi}_i^2 <$

$\mathbf{1}$, the series above is absolute convergent. Then,

$$
\begin{aligned}
&\mathsf{E}\left[g(x)^2\right] \\
&= 2^k \sum_{i \ 0} \widehat{\phi}_i^{\,2} \sum_{w 2 f \ 1 g^k} \prod_{l=1}^{k} w_l \left(\frac{\sum_{l=1}^{k} w_l}{k} \right)^i \\
&= 2^k \sum_{i \ 0} \frac{\widehat{\phi}_i^{\,2}}{k^i} \sum_{w 2 f \ 1 g^k} \prod_{l=1}^{k} w_l \sum_{i_1 + \ + i_k = i} \prod_{l=1}^{k} w_l^{i_l} \binom{i}{i_1, \ldots, i_k} \\
&= 2^k \sum_{i \ 0} \frac{\widehat{\phi}_i^{\,2}}{k^i} \sum_{i_1 + \ + i_k = i} \binom{i}{i_1, \ldots . i_k} \sum_{w 2 f \ 1 g^k} \prod_{l=1}^{k} w_l^{i_l + 1} \\
&= 2^k \sum_{i \ 0} \frac{\widehat{\phi}_i^{\,2}}{k^i} \sum_{i_1 + \ + i_k = i} \binom{i}{i_1, \ldots, i_k} \prod_{l=1}^{k} \left[1^{i_l + 1} + (\ \ 1)^{i_l + 1} \right] \\
&= 4^k \sum_{i \ 0} \frac{\widehat{\phi}_i^{\,2}}{k^i} \sum_{\substack{i_1 + \ + i_k = i \\ i_1, \ldots, i_k \text{ are odd}}} \binom{i}{i_1, \ldots, i_k}
\end{aligned}
$$

since we consider all distinct monomials in $\left(\sum_{l=1}^{k} w_l \right)^i$. Note that $\sum_{\substack{i_1 + \ + i_k = i \\ i_1, \ldots, i_k \text{ are odd}}} \binom{i}{i_1, \ldots, i_k}$ is always non-negative and is positive iff $i \quad k$ and $i \quad k \pmod 2$. □

A.1.1 ReLU Activation

The goal of this section is to give a lower-bound of $\mathsf{k} f \mathsf{k}$ for $\phi = \mathrm{ReLU}$ under the standard Gaussian distribution $\mathsf{N}(0, I)$. To this end, we prove an anti-concentration for g. We first give a lower bound on $\mathsf{k} g \mathsf{k}$ based on the Hermite coefficients of ϕ. If g were bounded, this alone would imply anti-concentration as in Section A.1.2. But since it is not, we first introduce g^T, where all activations are truncated at some T. We pick T large enough that g and g^T behave almost identically over $\mathsf{N}(0, I)$. We then show a lower bound on $\left\| g^T \right\|$, translate that into an anticoncentration result for g^T, and finally into one for g.

Let $T > 0$ be some constant to be determined later. Let

$$\mathrm{ReLU}^T(x) = \min(\mathrm{ReLU}(x), T)$$

and

$$g^T(x) = \sum_{w \in f \quad 1g^k} \chi(w)\,\mathrm{ReLU}^T\left(\frac{x \not{p} w}{k}\right).$$

The following lemma from [GKK19] describes the Hermite coefficients of ReLU.

Lemma A.1.2.

$$\mathrm{ReLU}(x) = \sum_{i=0}^{1} c_i \tilde{H}_i(x)$$

where

$$c_0 = \sqrt{\frac{1}{2\pi}}, \quad c_1 = \frac{1}{2},$$
$$c_{2i \quad 1} = 0, \quad c_{2i} = \frac{H_{2i}(0) + 2iH_{2i \quad 2}(0)}{\sqrt{2\pi(2i)!}} \quad \textit{for } i \quad 2.$$

In particular, $c_{2i}^2 = \Theta(i^{\quad 2.5})$.

We can now derive a lower bound on the norm of g.

Lemma A.1.3. *When k is even,*

$$\|g\| = \Omega\left(\left(\frac{4}{e}\right)^{(\frac{1}{2}+o(1))k}\right).$$

Proof. Due to Lemma A.1.1,

$$\mathsf{E}\left[g(x)^2\right] = 4^k \sum_{i \quad 0} \frac{c_i^2}{k^i} \sum_{\substack{i_1 + \quad + i_k = i \\ i_1,\ldots,i_k, \text{ are odd}}} \binom{i}{i_1, \ldots, i_k}$$
$$\frac{4^k c_k^2}{k^k} \sum_{\substack{i_1 + \quad + i_k = k \\ i_1,\ldots,i_k, \text{ are odd}}} \binom{k}{i_1, \ldots, i_k}$$
$$\frac{4^k c_k^2 k!}{k^k}.$$

The lemma then follows by the Stirling's approximation,

$$n! \quad \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

and the bound on the Hermite coefficients,

$$c_k^2 = \Theta(k^{-2.5}).$$

□

For the difference of $g(x)$ and $g^T(x)$, we have

Lemma A.1.4.

$$\|g - g^T\| \le 2^k e^{-\frac{T^2}{4}} \sqrt{T^2 + 1 - \frac{T}{\sqrt{2\pi}}}$$

Proof. Let $\mathrm{ReLU}_w(x)$ be shorthand for $\mathrm{ReLU}(\frac{x-w}{k})$, and similarly ReLU_w^T. Observe that by the triangle inequality,

$$\begin{aligned}\|g - g^T\| &= \left\| \sum_{w \in \{-1,1\}^k} \chi(w) \left(\mathrm{ReLU}_w - \mathrm{ReLU}_w^T\right) \right\| \\ &\le \sum_{w \in \{-1,1\}^k} \left\|\mathrm{ReLU}_w - \mathrm{ReLU}_w^T\right\| \\ &= 2^k \left\|\mathrm{ReLU} - \mathrm{ReLU}^T\right\|_{N(0,1)},\end{aligned}$$

where the last equality holds because for any unit vector v and $x \sim N(0, I)$, $x \cdot v$ has the distribution $N(0, 1)$. Now,

$$\left\|\mathrm{ReLU} - \mathrm{ReLU}^T\right\|_{N(0,1)}^2 = \int_T^{\infty} (x - T)^2 p(x)\, dx,$$

where $p(x)$ is the probability density function of $N(0, 1)$. Note that $p'(x) = -xp(x)$. We have

$$\begin{aligned}\int_T^{\infty} x^2 p(x) dx &= \int_T^{\infty} -x\, d(p(x)) \\ &= -x\, p(x)\Big|_T^{\infty} + \int_T^{\infty} p(x) dx \qquad \text{(integration by parts)} \\ &= T\, p(T) + \Pr_{x \sim N(0,1)}[x > T], \\ \int_T^{\infty} x\, p(x) dx &= -p(x)\Big|_T^{\infty} = p(T), \\ \int_T^{\infty} p(x) dx &= \Pr_{x \sim N(0,1)}[x > T] \le e^{-\frac{T^2}{2}}.\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathsf{E}\left[g(x) \quad g^T(x)\right]^2 \\
&\quad 4^k \left[(T^2+1)_{x \ \mathsf{P}_{N(0,1)}}(x > T) \quad T\,p(T)\right] \\
&\quad 4^k\, e^{\ \frac{T^2}{2}} \left(T^2+1 \quad \not{p}\frac{T}{2\pi}\right).
\end{aligned}$$

□

Lemma A.1.5.

$$\mathsf{P}[g(x) \not{6}= g^T(x)] \quad 2^k\, e^{\ \frac{T^2}{2}}.$$

Proof. For any w *2 f* $1g^k$,

$$\begin{aligned}
&{}_{x \ \mathsf{P}_{N(0,I)}}\left[\mathrm{ReLU}(\tfrac{x}{\not{p}}\tfrac{w}{k}) \not{6}= \mathrm{ReLU}^T(\tfrac{x}{\not{p}}\tfrac{w}{k})\right] \\
&= {}_{t \ \mathsf{P}_{N(0,1)}}[t > T] \\
&\quad e^{\ \frac{T^2}{2}}.
\end{aligned}$$

The lemma follows by a union bound. □

Lemma A.1.6.

$$\mathsf{P}\left[\mathit{j}g(x)\mathit{j} \quad 1\right] = \Omega(\exp(\ \ \Theta(k))).$$

Proof. For large enough $T = \Omega(k)$, it holds from Lemmas A.1.3 and A.1.4 that

$$\left\|g^T\right\| = \Omega\left(\left(\frac{4}{e}\right)^{(\frac{1}{2}+o(1))k}\right).$$

Since $\left|g^T(x)\right| \quad T\,2^k$,

$$\left\|g^T\right\|^2 = \mathsf{E}[g^T(x)^2] \quad 1 \ \ \mathsf{P}[\mathit{j}g^T(x)\mathit{j} \quad 1] + (T2^k)^2 \ \ \mathsf{P}[\mathit{j}g^T(x)\mathit{j} \quad 1],$$

so that

$$\mathsf{P}\left[\left|g^T(x)\right| \quad 1\right] = \frac{\Omega\left(\left(\frac{4}{e}\right)^{(1+o(1))k}\right) \quad 1}{(T\,2^k)^2} = \Omega(\exp(\ \ \Theta(k))) \tag{A.1.1}$$

Using Eq. (A.1.1) with Lemma A.1.5,

$$\mathsf{P}\left[|g(x)| \quad 1\right] = \Omega(\exp(\quad \Theta(k)))$$

for large enough $T = \Omega(k)$. □

The lower bound on $\|f\|$ now follows easily.

Corollary A.1.7.

$$\|f\| = \Omega(\exp(\quad \Theta(k))).$$

Proof. Since $f = \psi \quad g$, from Lemma A.1.6 and the fact that ψ is odd and increasing, we have that

$$\begin{aligned}\|f\| \quad & |\psi(1)| \ \mathsf{P}[g(x) \quad 1] + |\psi(\quad 1)| \ \mathsf{P}[g(x) \quad 1] \\ & = \psi(1) \ \mathsf{P}[|g(x)| \quad 1] \\ & = \Omega(\exp(\quad \Theta(k))).\end{aligned}$$

□

A.1.2 Sigmoid Activation

Here we consider g and f with $\phi(x) = \sigma(x) = \frac{1}{1+e^{\ x}}$. For the asymptotic bound of Hermite polynomial coefficients, we need the following theorem from [Boy84].

Theorem A.1.8. *For a function $f(z)$ whose convergence is limited by simple poles at the roots of $z^2 = \quad \gamma^2$ with residue R, the non-zero expansion coe cients $\{a_n\}$ of $f(z)$ as a series of normalized Hermite functions have magnitudes asymptotically given by*

$$|a_n| \quad 2^{\frac{5}{4}} \pi^{\frac{1}{2}} R\, n^{\ \frac{1}{4}} e^{\ \gamma(2n+1)^{\frac{1}{2}}},$$

Here the normalized Hermite function $\{\psi_n(x)\}_{n\in\mathbb{N}}$ is de ned by

$$\psi_n(z) = e^{\ \frac{z^2}{2}} \pi^{\ \frac{1}{4}} \tilde{H}_n(\sqrt{2}z).$$

Applying this to $f(x) = e^{\frac{x^2}{2}}\sigma(\sqrt{2}x)$ and translating the Hermite coefficients for the series in terms of Hermite functions to those in terms of Hermite polynomials, we have

Lemma A.1.9.

$$\sigma(x) = \sum_{i=0}^{1} c_i \tilde{H}_i(x),$$

where $c_0 = 0.5, c_{2i} = 0$ *for* $i \quad 1$ *and all non-zero odd terms satis es*

$$c_{2i\ 1} = e^{(\sqrt{i})}.$$

Corollary A.1.10. *There is an in nite increasing sequence* $fk_ig_{i2\mathbb{N}}$ *such that* k_i*'s are all odd and*

$$c_{k_i} = e^{(\sqrt{k_i})}.$$

Proof. It follows simply from the fact that σ is not a polynomial and there should be infinitely many non-zero terms in $fc_kg_{k2\mathbb{N}}$. □

Remark A.1.11. Experimental evidence strongly indicates that in fact all odd Hermite coefficients of sigmoid are nonzero and decay as above, but this is laborious to formally establish. So we state our norm lower bound only for $k\ 2\ fk_ig_{i2\mathbb{N}}$ (and the associated $n\ 2\ f2^{k_i}g_{i2\mathbb{N}}$, since we end up taking $k = \log n$). Since this is nevertheless an infinite sequence, it still establishes that no better asymptotic bound holds.

Similar to Lemma A.1.3, we can derive a lower bound of kgk for some k's.

Lemma A.1.12. *For* $k\ 2\ fk_ig_{i2\mathbb{N}}$*,*

$$kg(x)k = \Omega\left(\left(\frac{4}{e}\right)^{(\frac{1}{2}+o(1))k}\right).$$

Proof. Due to Lemma A.1.1,

$$\mathsf{E}\left[g(x)^2\right] = 4^k \sum_{i\ 0} \frac{c_i^2}{k^i} \sum_{\substack{i_1+\ \ +i_k=i \\ i_1,\ \ ,i_k,\ \text{are odd}}} \binom{i}{i_1\quad i_k}$$
$$\frac{4^k c_k^2}{k^k} \sum_{\substack{i_1+\ \ +i_k=k \\ i_1,\ \ ,i_k,\ \text{are odd}}} \binom{k}{i_1\quad i_k}$$
$$\frac{4^k c_k^2 k!}{k^k}.$$

Using Stirling's approximation,

$$n! \quad \sqrt{2\pi n}\left(\frac{n}{e}\right)^n,$$

and Corollary A.1.10,

$$c_k = e \quad (\sqrt{k}),$$

we obtain

$$\mathsf{E}\left[g(x)^2\right] = \Omega\left(\frac{4^k\sqrt{2\pi k}}{k^k}\left(\frac{k}{e}\right)^k e \quad (\sqrt{k})\right)$$

and hence

$$\mathsf{E}\left[g(x)^2\right] = \Omega\left(\left(\frac{4}{e}\right)^{(1+o(1))k}\right).$$

□

Lemma A.1.13. *For k 2 $\{k_i\}_{i2\mathbb{N}}$,*

$$\mathsf{P}\left(|g(x)| \quad 1\right) = \Omega(\exp(\quad \Theta(k))).$$

Proof. Since $|g(x)| \quad 2^k$,

$$\|g\|^2 = \mathsf{E}[g(x)^2] \quad 1 \quad \mathsf{P}[|g(x)| \quad 1] + (2^k)^2 \quad \mathsf{P}[|g(x)| \quad 1],$$

and so

$$\mathsf{P}\left(|g(x)| \quad 1\right) = \frac{\Omega\left(\left(\frac{4}{e}\right)^{(1+o(1))k}\right) \quad 1}{(2^k)^2}.$$

The lemma then follows. □

Using the same argument as Corollary A.1.7, we have the following bound.

Corollary A.1.14.

$$\|f\| = \Omega(\exp(-\Theta(k))).$$

A.1.3 General activations

It is not hard to see that the norm analysis of ReLU and sigmoid extends to any activation function for which a suitable lower bound on the Hermite coefficients holds, and which is either bounded or grows at a polynomial rate, so that under the standard Gaussian it behaves essentially identically to its truncated form. In particular, a lower bound of α^{-j} for any constant $\alpha < 4/e$ on the j^{th} Hermite coefficient suffices to give $\|g\| \geq \exp(\Theta(k))$, by the same argument as in Lemma A.1.3 and Lemma A.1.12. This then suffices to give $\|f\| \geq \exp(-\Theta(k))$, as above.

In fact, even a very weak lower bound on $\|f\|$ yields *some* superpolynomial bound on learning. Suppose we only had $\|f\| \geq 1/\exp(\exp(\Theta(k)))$, for instance. Then we can take $k = \log\log n$ and have $\|f\| \geq 1/\operatorname{poly}(n)$ and still obtain a lower bound of $n^{\log\log n} = n^{\omega(1)}$ (see Theorem 2.3.9). Any lower bound on $\|f\|$ will be a function only of k, so a similar argument applies.

A.2 SQ lower bound for real-valued functions proof

We give a self-contained variant of the elegant proof of [Szö09] for the reader's convenience. For simplicity, we include the 0 function in our class C — this can only negligibly change the SDA, and it makes the core argument cleaner.

Theorem A.2.1. *Let D be a distribution on X, and let C be a real-valued concept class over a domain X such that $0 \in C$, and $\|c\|_D > \epsilon$ for all $c \in C, c \neq 0$. Consider any SQ learner that is allowed to make only inner product queries to an SQ oracle for the labeled distribution D_c for some unknown $c \in C$. Let $d = \mathrm{SDA}_D(C, \gamma)$. Then any such SQ learner needs at least $d/2$ queries of tolerance $\sqrt{\gamma}$ to learn C up to L_2*

error ϵ.

Proof. Consider the adversarial strategy where we respond to every query $h : X \to \mathbb{R}$ ($\|h\|_D \leq 1$) with 0. This corresponds to the true expectation if the target were the 0 function. By the norm lower bound, outputting any other c would then mean L_2 error greater than ϵ. Thus we must rule out all other $c \in C$.

Let $\tau = \epsilon\sqrt{\gamma}$. If h_k is the k^{th} query, let $S_k = \{c \in C : |\langle c, h_k\rangle_D| > \tau\}$ be the functions ruled out by our response of 0. (A similar argument will hold for $S_k^0 = \{c \in C : |\langle c, h_k\rangle_D| < \tau\}$.) Let $\Phi = |\langle h_k, \sum_{c \in S_k} c\rangle_D|$. We claim that $|S_k| \leq |C|/d$. Suppose not. Then $\rho_D(S_k) \geq \gamma$ by Definition 2.2.4, and

$$\begin{aligned}\Phi &\leq \|h_k\|_D \left\|\sum_{c \in S_k} c\right\|_D \\ &\leq \sqrt{\sum_{c,c' \in S_k} \langle c, c'\rangle_D} \\ &= \sqrt{|S_k|^2 \rho_D(S_k)} \\ &\leq \epsilon\sqrt{\gamma}|S_k|,\end{aligned}$$

contradicting the fact that $\Phi > |S_k|\tau$ by definition of S_k.

Similarly $|S_k^0| = |\{c \in C : |\langle c, h_k\rangle_D| < \tau\}| \leq |C|/d$. Thus we rule out at most a $2/d$ fraction of functions with each query, and hence need at least $d/2$ queries to rule out all other possibilities. □

Appendix B: Hardness of Noise-Free Learning for Two-Hidden-Layer Neural Networks

B.1 Barriers for constructing N_3

We briefly discuss why one natural approach to constructing N_3 satisfying the ideal properties in Eq. (3.3.2) ultimately requires two hidden layers rather than one, unlike the construction we give in Sections 3.3.2 and 3.3.3.

The most straightforward way to ensure that a function of $s_1, \ldots, s_d, t$ vanishes whenever there exists j for which $s_j = 1$ would be to threshold on $\sum s_j$, e.g. by taking ReLU(1 $\sum_j s_j$). While this function is a one-hidden-layer ReLU network, it is unclear how to modify it to satisfy the remaining desiderata in (3.3.2) while preserving the fact that it has only one hidden layer. We note that [DV21] takes this approach of thresholding on $\sum_j s_j$ but uses two hidden layers.

Here we informally argue that such an approach inherently requires an extra hidden layer. That is, we argue that no function $N : \mathsf{R}^2 \; ! \; \mathsf{R}$ that takes as inputs s , $\sum_j s_j$ and t and satisfies (3.3.2) can be implemented as a one-hidden-layer network. Concretely, $N(s,t)$ must vanish whenever s 1 or t 2 Znf0g. Any function computed by a one-hidden-layer ReLU network of the form (s,t) 7! $\sum_i \mathrm{ReLU}(a_i s + b_i t$ $c_i)$, unless if it is affine linear, must in general be nowhere smooth (i.e. have a discontinuous gradient) along the entire line where a particular neuron of the network vanishes. In our example, these are the lines f$(s,t) : a_i s + b_i t = c_i$g. But this means that such a line cannot intersect the region f$(s,t) : s$ 1g, as otherwise it would be zero (hence smooth) on an infinite segment of the line. This can only happen if $b_i = 0$, i.e. none of the neurons of N depend on t. Such a network clearly cannot satisfy (3.3.2).

B.2 Supporting lemmas for Section 3.3

Lemma B.2.1. *For any* $0 \quad S < m \quad d$,

$$\sum_{i=S}^{m} (\quad 1)^{m\ i} \binom{d \quad i \quad 2}{d \quad m \quad 2} \binom{d \quad m \quad 1}{\quad i \quad S} = 0. \tag{B.2.1}$$

Proof. We will show that for any integers $j \quad \ell \quad 1$,

$$\sum_{k=0}^{\ell} (\quad 1)^{k} \binom{j \quad k}{\ell \quad 1} \binom{\ell}{k} = 0. \tag{B.2.2}$$

We would like to substitute $\ell = d \quad m \quad 1$ and $j = d \quad 2 \quad S$. Note that this is valid as we can assume without loss of generality that $d \quad m \quad 1 \quad 1$ (otherwise $\binom{d\ m\ 1}{i\ S} = 0$ on the right-hand side of (B.2.1)), and $j \quad \ell$ by our assumption that $S < m$. We conclude the identity

$$0 = \sum_{k=0}^{d\ m\ 1} (\quad 1)^{k} \binom{d \quad 2 \quad S \quad k}{d \quad m \quad 2} \binom{d \quad m \quad 1}{\quad k} = \sum_{i=S}^{d\ m\ 1+S} (\quad 1)^{i\ S} \binom{d \quad i \quad 2}{d \quad m \quad 2} \binom{d \quad m \quad 1}{\quad i \quad S}, \tag{B.2.3}$$

where the second step is by the change of variable $i = k + S$. If $d \quad m \quad 1 + S \quad m$, then note that all summands $m < i \quad d \quad m \quad 1 + S$ vanish because in that case $d \quad i \quad 2 < d \quad m \quad 2$ and so $\binom{d\ i\ 2}{d\ m\ 2} = 0$. If $d \quad m \quad 1 + S < m$, then note that all summands $d \quad m \quad 1 + S < i \quad m$ vanish because in that case $d \quad m \quad 1 < i \quad S$ and so $\binom{d\ m\ 1}{i\ S} = 0$. We conclude that (B.2.3) is equal, up to a sign, to the left-hand side of (B.2.1), so we'd be done.

It remains to establish (B.2.2), which we do by following an argument due to [Ear19]. Observe that the left-hand side of (B.2.2) is simply counting via inclusion-exclusion the number of subsets of $\{1, \ldots, j\}$ of size $\ell \quad 1$ which contain $\{1, \ldots, \ell\}$. Indeed, the $k = 0$ summand counts all subsets of size $\ell \quad 1$. The $k = 1$ summands subtract out the contribution, for every $1 \quad x \quad \ell$, from the subsets of size $\ell \quad 1$ which contain x. The $k = 2$ summands add back the contribution, for every distinct $1 \quad x < y \quad \ell$, from the subsets of size $\ell \quad 1$ which contain both of x, y, etc. □

Lemma B.2.2. *For any integers* $m \quad 3$ *and* a *2* $f0, 1, 2g$*,*

$$\sum_{i=1}^{m}\sum_{j=1}^{m+1}{}^{i}(\quad 1)^{m\ i}\binom{d \quad i \quad j \quad 1}{m \quad i \quad j+1}\binom{d \quad 1}{i \quad 1}\quad j^{a} = \mathbb{1}[a=0] \tag{B.2.4}$$

$$\sum_{i=0}^{m}\sum_{j=1}^{m+1}{}^{i}(\quad 1)^{m\ i}\binom{d \quad i \quad j \quad 1}{m \quad i \quad j+1}\binom{d \quad 1}{\quad i}\quad j^{a} = 0 \tag{B.2.5}$$

Proof. By taking $\ell = i + j$, we can rewrite these sums as

$$S_{a,m}\ ,\ \sum_{\ell=2}^{m+1}\sum_{i=1}^{\ell\ 1}(\quad 1)^{m\ i}\binom{d \quad 1 \quad \ell}{m+1 \quad \ell}\binom{d \quad 1}{i \quad 1}(\ell \quad i)^{a} \tag{B.2.6}$$

$$T_{a,m}\ ,\ \sum_{\ell=1}^{m+1}\sum_{i=0}^{\ell\ 1}(\quad 1)^{m\ i}\binom{d \quad 1 \quad \ell}{m+1 \quad \ell}\binom{d \quad 1}{\quad i}(\ell \quad i)^{a} \tag{B.2.7}$$

We proceed by induction on m. The base cases follow from a direct calculation. By the change of variable $\ell^{0} = \ell \quad 1$, we can rewrite $S_{a,m+1}$ as

$$\sum_{\ell^{0}=1}^{m+1}\sum_{i=1}^{\ell^{0}}(\quad 1)^{m\ i}\binom{d \quad 1 \quad \ell^{0}}{m+1 \quad \ell^{0}}\binom{d \quad 1}{i \quad 1}(\ell^{0}+1 \quad i)^{a} \tag{B.2.8}$$

$$= \sum_{\ell^{0}=1}^{m+1}\sum_{i=1}^{\ell^{0}}(\quad 1)^{m\ i}\binom{d \quad 1 \quad \ell^{0}}{m+1 \quad \ell^{0}}\binom{d \quad 1}{i \quad 1}(\ell^{0} \quad i)^{a} \tag{B.2.9}$$

$$\sum_{\ell^{0}=1}^{m+1}\sum_{i=1}^{\ell^{0}}(\quad 1)^{m\ i}\binom{d \quad 1 \quad \ell^{0}}{m+1 \quad \ell^{0}}\binom{d \quad 1}{i \quad 1}\sum_{b=0}^{a\ 1}\binom{a}{b}(\ell^{0} \quad i)^{b} \tag{B.2.10}$$

Note that the first term on the right-hand side differs from $S_{a,m}$ only in the summands given by $1 \quad i = \ell' \quad m+1$, and those summands clearly vanish. We conclude that the first term on the right-hand side of (B.2.10) is exactly $S_{a,m}$. For the second term on the right-hand side of (B.2.10), the part coming from any $0 < b \quad a \quad 1$ is also zero, so we thus get

$$= S_{a,m} \quad \sum_{\ell'=1}^{m+1} \sum_{i=1}^{\ell'} (\quad 1)^m \quad i \binom{d \quad 1 \quad \ell'}{m+1 \quad \ell'} \binom{d \quad 1}{i \quad 1} \tag{B.2.11}$$

$$= S_{a,m} \quad S_{0,m} \quad \sum_{\ell'=1}^{m+1} (\quad 1)^m \quad \ell' \binom{d \quad 1 \quad \ell'}{m+1 \quad \ell'} \binom{d \quad 1}{\ell' \quad 1} \tag{B.2.12}$$

$$= S_{a,m} \quad 1 \quad \sum_{\ell'=1}^{m+1} (\quad 1)^m \quad \ell' \binom{d \quad 1 \quad \ell'}{m+1 \quad \ell'} \binom{d \quad 1}{\ell' \quad 1} \tag{B.2.13}$$

$$= S_{a,m} = \mathbb{1}[a = 0], \tag{B.2.14}$$

where the penultimate step follows e.g. by applying the identity in [PSP17]. This completes the induction for $S_{a,m}$.

For $T_{a,m}$, note that by the change of variable $i' = i + 1$,

$$T_{a,m} = \quad \sum_{\ell=1}^{m+1} \sum_{i'=1}^{\ell} (\quad 1)^m \quad i' \binom{d \quad 1 \quad \ell}{m+1 \quad \ell} \binom{d \quad 1}{i' \quad 1} (\ell \quad i' + 1)^a \tag{B.2.15}$$

$$= \quad \sum_{\ell=2}^{m+1} \sum_{i'=1}^{\ell \quad 1} (\quad 1)^m \quad i' \binom{d \quad 1 \quad \ell}{m+1 \quad \ell} \binom{d \quad 1}{i' \quad 1} (\ell \quad i' + 1)^a \quad \sum_{\ell=1}^{m+1} (\quad 1)^m \quad \ell \binom{d \quad 1 \quad \ell}{m+1 \quad \ell} \binom{d \quad 1}{\ell \quad 1} \tag{B.2.16}$$

$$= \quad \sum_{b=0}^{a} \binom{a}{b} S_{b,m} + 1 = 0, \tag{B.2.17}$$

where in the second step we pulled out the summands corresponding to $i' = \ell$, in the third step we used (B.2.14), and in the last step we used that for $m \quad 3$, $S_{b,m} = \mathbb{1}[b \neq 0]$ for $0 \quad b \quad 2$. □

B.3 SQ lower bound for the LWR functions

Here we prove an SQ lower bound for the LWR functions (Theorem 3.4.5) using a general formulation in terms of pairwise independent function families. To

our knowledge, this particular formulation has not appeared explicitly before in the literature, and was communicated to us by Bogdanov [Bog21]. A variant of this argument may be found in [BR17, §7.7].

Definition B.3.1. Let $\mathcal{C}$ be a function family mapping X to Y, and let D be a distribution on X. We call $\mathcal{C}$ an $(1 - \eta)$-pairwise independent function family if with probability $1 - \eta$ over the choice of x, x' drawn independently from D, the distribution of $(f(x), f(x'))$ for f drawn uniformly at random from $\mathcal{C}$ is the product distribution $\mathrm{Unif}(Y) \times \mathrm{Unif}(Y)$.

Lemma B.3.2. *Fix security parameter n and moduli p, q. The $\mathsf{LWR}_{n,p,q}$ function class $\mathcal{C}_{\mathsf{LWR}} = \{f_w \mid w \in \mathbb{Z}_q^n\}$ is $(1 - \frac{2}{q^n - 1})$-pairwise independent with respect to $\mathrm{Unif}(\mathbb{Z}_q^n)$.*

Proof. This follows from the simple observation that whenever $x, x' \in \mathbb{Z}_q^n$ are linearly independent, the pair $(w \cdot x \bmod q, w \cdot x' \bmod q)$ for $w \leftarrow \mathrm{Unif}\{\mathbb{Z}_q^n\}$ is distributed as $\mathrm{Unif}(\mathbb{Z}_q) \times \mathrm{Unif}(\mathbb{Z}_q)$. For such x, x', $(f_w(x), f_w(x')) = (\frac{1}{p} \cdot (w \cdot x \bmod q) \cdot e_p), \frac{1}{p} \cdot (w \cdot x' \bmod q) \cdot e_p)$ for $f_w \leftarrow \mathrm{Unif}(\mathcal{C}_{\mathsf{LWR}})$ is distributed as $\mathrm{Unif}(\mathbb{Z}_p/p) \times \mathrm{Unif}(\mathbb{Z}_p/p)$. The probability that $x, x' \leftarrow \mathrm{Unif}(\mathbb{Z}_q^n)$ are linearly dependent is at most

$$\mathbb{P}[x = 0] + \mathbb{P}[x \neq 0] \, \mathbb{P}[x' \text{ is a multiple of } x] \leq \frac{1}{q^n} + \frac{q}{q^n} \leq \frac{2}{q^n - 1}.$$

□

We can now prove full SQ lower bounds for any $(1 - \eta)$-pairwise independent function family as follows.

Lemma B.3.3. *Let $\mathcal{C}$ mapping X to Y be a $(1 - \eta)$-pairwise independent function family w.r.t. a distribution D on X. Let $\phi : X \times Y \to [-1, 1]$ be any bounded query function. Then*

$$\mathop{\mathrm{Var}}_{f \leftarrow \mathrm{Unif}(\mathcal{C})} \mathop{\mathbb{E}}_{x \leftarrow D} [\phi(x, f(x))] \leq 2\eta.$$

Proof. Denote $\mathsf{E}_{x \quad D}[\phi(x, f(x))]$ by $\phi[f]$. By some algebraic manipulations (with all subscripts denoting independent draws),

$$\begin{aligned}
{}_{f \quad \mathsf{Unif}(\mathsf{C})}\operatorname{Var}[\phi[f]] &= \mathop{\mathsf{E}}_{f}\left[\phi[f]^2\right] \quad \left(\mathop{\mathsf{E}}_{f}[\phi[f]]\right)^2 && \text{(B.3.1)}\\
&= \mathop{\mathsf{E}}_{f}[\phi[f]\phi[f]] \quad \mathop{\mathsf{E}}_{f}[\phi[f]] \mathop{\mathsf{E}}_{f^0}[\phi[f^0]] && \text{(B.3.2)}\\
&= \mathop{\mathsf{E}}_{f,f^0}\left[\mathop{\mathsf{E}}_{x}[\phi(x, f(x))] \mathop{\mathsf{E}}_{x^0}[\phi(x^0, f(x^0))] \quad \mathop{\mathsf{E}}_{x}[\phi(x, f(x))] \mathop{\mathsf{E}}_{x^0}[\phi(x^0, f^0(x^0))]\right] && \text{(B.3.3)}\\
&= \mathop{\mathsf{E}}_{x,x^0} \mathop{\mathsf{E}}_{f,f^0}[\phi(x, f(x))\phi(x^0, f(x^0)) \quad \phi(x, f(x))\phi(x^0, f^0(x^0))]\,. && \text{(B.3.4)}
\end{aligned}$$

By $(1 \quad \eta)$-pairwise independence of C, the inner expectation vanishes with probability $1 \quad \eta$ over the choice of $x, x^0 \quad D$, and is at most 2 otherwise. This gives the claim. □

Theorem B.3.4. *Let C mapping X to Y be a $(1 \quad \eta)$-pairwise independent function family w.r.t. a distribution D on X. For any f 2 C, let D_f denote the distribution of $(x, f(x))$ where $x \quad D$. Let $D_{\mathsf{Unif}(\mathsf{C})}$ denote the distribution of (x, y) where $x \quad D$ and $y = f(x)$ for $f \quad \mathrm{Unif}(\mathsf{C})$ (this can be thought of as essentially $D \quad \mathrm{Unif}(Y)$). Any SQ learner able to distinguish the labeled distribution D_f for an unknown f 2 C from the randomly labeled distribution $D_{\mathsf{Unif}(\mathsf{C})}$ using bounded queries of tolerance τ requires at least $\frac{\tau^2}{2\eta}$ such queries.*

Proof. Let $\phi : X \quad Y \; ! \; [\ 1, 1]$ be any query made by the learner. For any f 2 C, let $\phi[f]$ denote $\mathsf{E}_{x \quad D}[\phi(x, f(x))] = \mathsf{E}_{(x,y) \quad D_f}[\phi(x, y)]$. Consider the adversarial strategy where the SQ oracle responds to this query with $\overline{\phi} = \mathsf{E}_{f \quad \mathsf{Unif}(\mathsf{C})}\,\phi[f] = \mathsf{E}_{(x,y) \quad D_{\mathsf{Unif}(\mathsf{C})}}[\phi(x, y)]$. By Chebyshev's inequality and Lemma B.3.3,

$$\mathop{\mathsf{P}}_{f \quad \mathsf{C}}\left[\left|\phi[f] \quad \overline{\phi}\right| > \tau\right] \quad \frac{\mathrm{Var}_{f \quad \mathsf{Unif}(\mathsf{C})}\left[\phi[f]\right]}{\tau^2} \quad \frac{2\eta}{\tau^2}.$$

So each such query only allows the learner to rule out at most a $\frac{2\eta}{\tau^2}$ fraction of C. Thus to distinguish D_f from $D_{\mathsf{Unif}(\mathsf{C})}$, the learner requires at least $\frac{\tau^2}{2\eta}$ queries. □

Theorem 3.4.5 now follows easily as a corollary.

Proof of Theorem 3.4.5. It is not hard to see that learning $\mathsf{C}_{\mathsf{LWR}}$ up to squared loss $1/16$ certainly suffices to solve the distinguishing problem in Theorem B.3.4. The claim now follows by Lemma B.3.2. □

Remark B.3.5. We remark that the argument in this section, specialized to the $q = 2$ case, recovers the traditional SQ lower bound for parities (Theorem 3.4.3) without appealing to any notion of statistical dimension.

Appendix C: Statistical-Query Lower Bounds via Functional Gradients

C.1 SQ lower bound subtleties

C.1.1 Relationships between parameters

When formally stating SQ lower bounds on learning p-concepts in terms of the statistical dimension, there are some subtleties to keep in mind. These have to do with the relationships between the query tolerance, the desired final error, and the norms of the functions in the class. Let us say our queries are of tolerance τ, the final desired L^2 error $\|f - f'\|$ is ϵ (which corresponds to $L(f) - L(f') \geq \epsilon^2$; see Eq. (4.3.1)), and that the functions in $\mathcal{C}$ satisfy $\|f\| \leq \beta$ for all $f \in \mathcal{C}$. Then

1. We must have $\tau < \epsilon$. To see why, first note that for any query ϕ and two functions $f, g \in \mathcal{C}$, a calculation shows that $|\mathbf{E}_{D_f}[\phi] - \mathbf{E}_{D_g}[\phi]| = |\langle f - g, \tilde{\phi} \rangle| \leq \|f - g\|$, where $\tilde{\phi}(x) = (\phi(x,1) - \phi(x,-1))/2$. Thus if one has a function f' such that $\epsilon < \|f - f'\| < \tau$, then no query of tolerance τ can tell them apart, but f' is not ϵ-close to the target f.

2. If $\epsilon \geq \beta$, a lower bound might not be possible. This is because the 0 function trivially achieves L^2 error $\|0 - f\| = \|f\|$. Imposing $\epsilon < \beta$ is sufficient to rule this out.

3. We cannot arbitrarily rescale the p-concepts to increase β since the functions must remain Boolean p-concepts. Rescaling would also increase the description length of the functions.

The lower bound in Theorem 4.2.2 (from [GGJ+20]) is proved by reducing a distinguishing problem to a learning problem. For technical reasons, we end up requiring $\tau \leq \epsilon^2$, $\epsilon \leq \beta/3$ for this reduction to go through. The points above show that these requirements are essentially necessary.

C.1.2 The dependence of the query lower bound on the error ϵ and the tolerance τ

The relationship between our query lower bounds, the desired error ϵ, and the tolerance τ may seem a little unusual at first sight, especially the fact that the lower bounds seem to grow weaker as ϵ grows smaller. We make some clarifying remarks here.

Fundamentally, all SQ lower bounds are bounds on how many queries it takes to distinguish certain distributions from others. When discussing a concept class C, the distributions in question are the labeled distributions corresponding to concepts in the class. Learning C is hard exactly insofar as it allows us to distinguish different labeled distributions arising from C. Many works in the SQ literature have this structure, but we will refer to [GGJ+20] for formal statements.

Formally, the distinguishing problem we consider ([GGJ+20, Definition 4.2]) is that of distinguishing the labeled distribution D_c arising from an unknown $c \in \mathsf{C}$ from the reference distribution $D_0 = D \quad \mathrm{Unif}\{-1, 1\}$, using queries of tolerance at least τ.

There are two crucial points to keep in mind here:

1. The distinguishing problem is a fundamentally information theoretic problem, and its difficulty scales only with τ. In particular, using queries of tolerance τ, we need at least $\mathrm{SDA}(\mathsf{C}, \tau^2)$ queries. This bound increases with τ; in fact it often scales as $|\mathsf{C}|\tau^2$ (see ([GGJ+20, Theorem 4.5 and Lemma 2.6]).

2. The problem of learning up C to error ϵ is hard exactly insofar as it allows us to solve the distinguishing problem (see [GGJ+20, Lemma 4.4]).

An important consequence is that for fixed τ, the query lower bound does not technically grow as a function of the error ϵ: it applies uniformly for all ϵ small enough that it allows the learner to solve the distinguishing problem. In other words,

there is a certain "threshold" ϵ_0 such that for all ϵ ϵ_0, the same query lower bound holds. As noted in point (3) of the previous subsection, this threshold can be taken to be $\beta/3$, where β is such that kck β for all c 2 C.

But at the same time, as noted in point (1) in the previous subsection, it is necessary that $\tau < \epsilon$ (and for the reduction it suffices to have τ ϵ^2). If τ ϵ, learning up to error ϵ is simply impossible.

With all this in mind, we can now answer the question of why our lower bounds seem to grow weaker as ϵ grows smaller: it is essentially because τ grows smaller as well, so that we get a series of incomparable (though still exponential) bounds due to the tradeoffs between query complexity, τ, and ϵ.

C.2 Bounding the function norms of the [DKKZ20] construction

We shall consider the following slight rescaling of the functions of [DKKZ20]. For activation functions $\psi, \phi : \mathsf{R}$! R, we have $g, f : \mathsf{R}^2$! R defined as follows.

$$g(x) = \frac{1}{2m}\sum_{i=1}^{2m}(\quad 1)^i \phi\left(x_1 \cos\frac{i\pi}{m} + x_2 \sin\frac{i\pi}{m}\right) = \frac{1}{2m}\sum_{i=1}^{2m}(\quad 1)^i \phi\left(x \quad w_i\right)$$

$$f(x) = \psi(g(x)),$$

where $w_i = (\cos\frac{i\pi}{m}, \sin\frac{i\pi}{m})$. The number of hidden units is $k = 2m$. We will assume that m is even.

The hard functions from R^n ! R are then given by $f_A(x) = f(Ax)$ for certain matrices A 2 $\mathsf{R}^{2\ d}$ with $AA^T = I_2$. For x $N(0, I_d)$, Ax has the distribution $N(0, I_2)$. So for the purposes of the norm calculation, and hence throughout this section, we will work directly with $N(0, I_2)$. We will start by considering the norm of g. This can then be used to control the norm of f via arguments similar to those in [GGJ+20].

Lemma C.2.1. *Let $g : \mathbb{R}^2 \to \mathbb{R}$ be as defined above, and assume m is even. Assume the standard Hermite expansion of ϕ is given by $\phi = \sum_a \widehat{\phi}_a H_a$, where the H_a are the normalized probabilists' Hermite polynomials. Under $N(0, I_2)$,*

$$\|g\|^2 = \Omega\left(\sum_{\substack{a \\ a \text{ even}}}^{m} \widehat{\phi}_a^2 \sqrt{a}\right).$$

(For practical purposes, the asymptotic behavior of this expression is captured faithfully when we begin indexing from say $a = 100m$.)

Proof. We have

$$\begin{aligned}
\|g\|^2 = \mathbb{E}[g(x)^2] &= \frac{1}{4m^2}\sum_{i,j=1}^{2m}(-1)^i(-1)^j\,\mathbb{E}[\phi(x - w_i)\phi(x - w_j)] \\
&= \frac{1}{4m^2}\sum_{i,j=1}^{2m}(-1)^i(-1)^j\,\mathbb{E}\left[\left(\sum_a \widehat{\phi}_a H_a(x - w_i)\right)\left(\sum_b \phi_b H_b(x - w_j)\right)\right] \\
&= \frac{1}{4m^2}\sum_{i,j=1}^{2m}(-1)^i(-1)^j\left(\sum_a \widehat{\phi}_a^2\,\mathbb{E}[H_a(x - w_i)H_a(x - w_j)]\right).
\end{aligned}$$

Now because w_i, w_j are both unit vectors with $w_i \cdot w_j = \cos\frac{(i-j)\pi}{m}$, we have that $x - w_i$ and $x - w_j$ are both $N(0,1)$ with covariance $\cos\frac{(i-j)\pi}{m}$. Thus

$$\begin{aligned}
\|g\|^2 &= \frac{1}{4m^2}\sum_{i,j=1}^{2m}(-1)^{i+j}\left(\sum_a \widehat{\phi}_a^2 \cos^a\frac{(i-j)\pi}{m}\right) \\
&= \frac{1}{4m^2}\sum_{i,j=1}^{2m}(-1)^{i-j}\left(\sum_a \widehat{\phi}_a^2 \cos^a\frac{(i-j)\pi}{m}\right),
\end{aligned}$$

since $(-1)^{i+j} = (-1)^{i-j}$. Now, as we range over $i, j \in [2m]$, we see that $i - j = 0$ occurs $2m$ times, $i - j = 1$ occurs $2m - 1$ times, and more generally $i - j = t$ occurs $2m - |t|$ times. Since a term with $i - j = t$ is exactly the same as one with $i - j = -t$ (by the evenness of cos), we can say that for $t \neq 0$, $|i - j| = |t|$ occurs $2(2m - t)$ times.

Thus the expression above can be written as

$$\begin{aligned}
\|g\|^2 &= \frac{1}{4m^2}\left(2m\left(\sum_a \widehat{\phi}_a^2 \cos^a 0\right) + \sum_{t=1}^{2m-1} 2(2m-t)(-1)^t\left(\sum_a \widehat{\phi}_a^2 \cos^a \frac{t\pi}{m}\right)\right) \\
&= \frac{1}{4m^2}\sum_a \widehat{\phi}_a^2\left(2m + \sum_{t=1}^{2m-1} 2(2m-t)(-1)^t \cos^a \frac{t\pi}{m}\right) \\
&= \frac{1}{4m^2}\sum_a \widehat{\phi}_a^2 S(a,m), \qquad \text{(C.2.1)}
\end{aligned}$$

where

$$S(a,m) = 2m + \sum_{t=1}^{2m-1} 2(2m-t)(-1)^t \cos^a \frac{t\pi}{m}.$$

Now some algebraic manipulations are in order. By rewriting the index t as $2m-t$, we get that

$$\begin{aligned}
S(a,m) &= 2m + \sum_{t=1}^{2m-1} 2t(-1)^{2m-t} \cos^a \frac{(2m-t)\pi}{m} \\
&= 2m + \sum_{t=1}^{2m-1} 2t(-1)^t \cos^a \frac{t\pi}{m}.
\end{aligned}$$

Adding the two expressions for $S(a,m)$ and dividing by 2, we get

$$\begin{aligned}
S(a,m) &= 2m + \sum_{t=1}^{2m-1} 2m(-1)^t \cos^a \frac{t\pi}{m} \\
&= 2m \sum_{t=0}^{2m-1} (-1)^t \cos^a \frac{t\pi}{m}
\end{aligned}$$

This sum vanishes when a and m have different parities, i.e. if a is odd (recall that we assume m is even). For even a, we have

$$S(a,m) = 4m \sum_{t=0}^{m-1} (-1)^t \cos^a \frac{t\pi}{m}.$$

This is a trigonometric power sum with known closed form expressions. In particular,

Equation 3.4 from [DFGK17, §3] (after correcting a typo) tells us that

$$T(a,m) = \sum_{t=0}^{m-1} (-1)^t \cos^a \frac{t\pi}{m} = \begin{cases} 2^{-a} m \left(\sum_{p=1}^{\lfloor a/m \rfloor} \binom{a}{a/2 - pm/2} - \sum_{p=1}^{\lfloor a/2m \rfloor} \binom{a}{a/2 - pm} \right) & a \geq 2m \\ 2^{-a} m \sum_{p=1}^{\lfloor a/m \rfloor} \binom{a}{a/2 - pm/2} & m \leq a < 2m \\ 0 & a < m \end{cases}$$

$$= \begin{cases} 2^{-a} m \left(\sum_{\substack{p=1 \\ p \text{ odd}}}^{\lfloor a/m \rfloor} \binom{a}{a/2 - pm/2} \right) & a \geq m \\ 0 & a < m \end{cases}$$

To get a sense for the asymptotics as $a \to \infty$, we consider $a \geq m$ (say $a \geq 100m$). In this regime the sum of binomial coefficients in the sum above is seen to be $\Omega(2^a/\sqrt{a})$ (the $p = 1$ term alone contributes roughly $\binom{a}{a/2}$), and we get that $T(a,m) = \Omega(m/\sqrt{a})$.

This means $S(a,m) = 0$ for odd a and $S(a,m) = 4mT(a,m) = \Omega(m^2/\sqrt{a})$ for large, even a. Substituting this back into Eq. (C.2.1), we get that

$$\|g\|^2 = \Omega\left(\sum_{\substack{a \geq m \\ a \text{ even}}} \frac{\widehat{\phi}_a^2}{\sqrt{a}} \right).$$

□

We can now consider the special cases of $\phi = \mathrm{ReLU}$ and $\phi = \sigma$ (the standard sigmoid) that are of interest.

Corollary C.2.2. *Consider g instantiated with $\phi = \mathrm{ReLU}$. Then $\|g\| = \Omega(1/m)$.*

Proof. The Hermite coefficients of ReLU satisfy $\widehat{\phi}_a = \Theta(a^{-5/4})$ (Lemma C.3.1). Thus by Lemma C.2.1,

$$\|g\|^2 = \Omega\Big(\sum_{\substack{a \geq 100m \\ a \text{ even}}} a^{-3} \Big) = \Omega(1/m^2).$$

□

Corollary C.2.3. *Consider g instantiated with $\phi = \sigma$, the standard sigmoid. Then $\|g\| = e^{O(\sqrt{m})}$.*

Proof. The Hermite coefficients of σ asymptotically satisfy $\widehat{\phi}_a \sim e^{-C\sqrt{a}}$ [GGJ+20, §A.2] for some C. Thus by Lemma C.2.1,

$$\|g\|^2 = \Omega\Big(\sum_{\substack{a \\ a \text{ even}}}^{100m} \frac{e^{-\sqrt{a}}}{\sqrt{a}}\Big).$$

The result then follows by the following standard integral approximation:

$$\sum_{t=N}^{1} \frac{e^{-\sqrt{t}}}{\sqrt{t}} \qquad \int_N^1 \frac{e^{-\sqrt{t}}}{\sqrt{t}}\, dt = 2e^{-\sqrt{N}}.$$

□

We can now translate these into norm lower bounds on $f = \psi \circ g$. For us it suffices to consider $\psi = \tanh : \mathbb{R} \to [-1, 1]$, which is essentially the sigmoid centered at 0. The centering at 0 and the output range being $[-1, 1]$ is what is important to us, because we use f to capture the conditional mean function of a p-concept.

Lemma C.2.4. *Consider f instantiated with $\psi = \tanh$ and $\phi = \mathrm{ReLU}$. Then $\|f\| = \Omega(1/m^6)$.*

Proof. Ideally we would like to use the norm bound on g to obtain an anti-concentration inequality of the form $\mathbb{P}[|g(x)| > t]$, and then translate that into a norm lower bound for f, but this is not immediate because g is unbounded. So we introduce the function g^T, which is the same as g except with the truncated ReLU, $\mathrm{ReLU}^T(x) = \min(T, \mathrm{ReLU}(x))$ (T to be determined), in place of all standard ReLUs. Clearly $|g^T(x)| \le T$ for all x. It is also easy to see by a union bound that

$$\mathbb{P}[g(x) \neq g^T(x)] \le 2m \cdot \mathbb{P}_{t \sim N(0,1)}[\mathrm{ReLU}(t) \neq \mathrm{ReLU}^T(t)] \le 2me^{-T^2/2},$$

since each w_i is a unit vector.

Let $\mathrm{ReLU}_w(x)$ be shorthand for $\mathrm{ReLU}(x \quad w)$, and similarly ReLU_w^T. Observe first that

$$\begin{aligned}\|g \quad g^T\| &= \frac{1}{2m} k \; k \sum_{i=1}^{2m} (\quad 1)^i (\mathrm{ReLU}_{w_i} \quad \mathrm{ReLU}_{w_i}^T) \\ &\quad \frac{1}{2m} \sum_{i=1}^{2m} \|\mathrm{ReLU}_{w_i} \quad \mathrm{ReLU}_{w_i}^T)\| \\ &= \|\mathrm{ReLU} \quad \mathrm{ReLU}^T\|_{N(0,1)} \\ &\quad \sqrt{e^{\frac{T^2}{2}} \left(T^2 + 1 \quad p\frac{T}{2\pi}\right)}\end{aligned}$$

where the third equality again uses the fact the w_i are unit vectors, and the last inequality is Lemma C.2.5. By picking $T = \Theta(m)$, this coupled with the fact that $kgk = \Omega(1/m)$ (Corollary C.2.2) tells us that $\|g^T\| = \Omega(1/m)$ as well.

This bound on $\|g^T\|$ yields an anti-concentration inequality for g^T as follows:

$$kg^T k^2 = \mathsf{E}[g^T(x)^2] \quad t^2 \, \mathsf{P}[jg^T(x)j \quad t] + T^2 \, \mathsf{P}[jg^T(x)j > t] = t^2 + (T^2 \quad t^2) \, \mathsf{P}[jg^T(x)j > t],$$

so that

$$\mathsf{P}[jg^T(x)j > t] \quad \frac{kg^T k^2 \quad t^2}{T^2 \quad t^2}.$$

Recall that $\mathsf{P}[g(x) \neq g^T(x)] \quad 2me^{\quad T^2/2}$, so

$$\mathsf{P}[jg(x)j > t] \quad \frac{kg^T k^2 \quad t^2}{T^2 \quad t^2} \quad 2me^{\quad T^2/2}.$$

Thus by taking $T = \Theta(m)$ and $t = \Theta(1/m)$, we get that

$$\mathsf{P}[jg(x)j > \Theta(1/m)] \quad \Omega(1/m^4).$$

Thus finally we have

$$kfk = \mathsf{E}[\tanh(g(x))^2] \quad \tanh^2(\Theta(1/m))\Omega(1/m^4) \quad \Omega(1/m^6),$$

since $\tanh(x) \quad x \quad x^3$ for small x (by its Taylor series). □

Lemma C.2.5 ([GGJ+20], Appendix A.1)**.** *For* $\mathrm{ReLU}^T(x) = \min(T, \mathrm{ReLU}(x))$,

$$\left\|\mathrm{ReLU} \quad \mathrm{ReLU}^T\right\|_{N(0,1)} \quad \sqrt{e^{\frac{T^2}{2}}\left(T^2+1 \quad \frac{T}{\sqrt{2\pi}}\right)}.$$

Proof. Let $p(t) = \frac{1}{\sqrt{2\pi}} e^{t^2/2}$ be the pdf of $N(0,1)$. Then

$$\begin{aligned}
\left\|\mathrm{ReLU} \quad \mathrm{ReLU}^T\right\|_{N(0,1)}^2 &= \mathop{\mathsf{E}}_{t \ N(0,1)}\left[\left(\mathrm{ReLU}(t) \quad \mathrm{ReLU}^T(t)\right)^2\right] \\
&= \int_T^1 (t \quad T)^2 p(t)\, dt \\
&= \int_T^1 t^2 p(t)\, dt \quad 2T\int_T^1 t p(t)\, dt + T^2 \int_T^1 p(t)\, dt
\end{aligned}$$

Noting that $p'(t) = \quad tp(t)$, we have

$$\begin{aligned}
\int_T^1 t^2 p(t)\, dt &= \int_T^1 \quad t\, d(p(t)) \\
&= \quad t\, p(x)\Big|_T^1 + \int_T^1 p(t)\, dt \qquad \text{(integration by parts)} \\
&= T\, p(T) + \mathop{\mathsf{P}}_{t \ N(0,1)}(t > T), \\
\int_T^1 t\, p(t)\, dt &= \quad p(t)\Big|_T^1 = p(T), \\
\int_T^1 p(t)\, dt &= \mathop{\mathsf{P}}_{t \ N(0,1)}(t > T) \quad e^{\frac{T^2}{2}}.
\end{aligned}$$

The claim follows by algebra. □

Lemma C.2.6. *Consider* f *instantiated with* $\psi = \tanh$ *and* $\phi = \sigma$*. Then* $\|f\| = e^{O(\sqrt{p}\overline{m})}$.

Proof. Here the same approach as above becomes considerably simpler since $|g(x)|$ 1 always. The norm bound on g yields the following anti-concentration inequality:

$$\mathsf{P}[|g(x)| > t] \quad \frac{\|g\|^2 \quad t^2}{1 \quad t^2}.$$

In our case, taking $t = e^{C\sqrt{p}\overline{m}}$ for sufficiently large C and using $\|g\| = e^{O(\sqrt{p}\overline{m})}$ (Corollary C.2.3) yields

$$\mathsf{P}[|g(x)| > e^{C\sqrt{p}\overline{m}}] = e^{O(\sqrt{p}\overline{m})}.$$

Thus

$$\|f\| = \mathbb{E}[\tanh(g(x))^2] \quad \tanh^2(e^{-C\sqrt{m}})e^{-O(\sqrt{m})} \quad e^{-O(\sqrt{m})},$$

since again $\tanh(x) \quad x \quad x^3$ for small x. □

C.3 Approximate degree of ReLUs and sigmoids

Here we give estimates for the δ-approximate degree of ReLUs and sigmoids under the standard Gaussian using bounds on their Hermite coefficients. Recall that we consider units $\phi(w \cdot x)$ with $\|w\|_2 \quad 1$. It is clear that for $\phi = \mathrm{ReLU}$ and $\phi = \sigma$, the norm only increases monotonically with $\|w\|_2$, so for the purposes of analysis it suffices to consider exactly $\|w\|_2 = 1$.

It is not hard to show that whenever w is a unit vector, the total-degree-d Hermite weight of $\phi(w \cdot x)$ as $x \quad \mathcal{N}(0, I_n)$ is the same as that of the univariate $\phi(t)$ as $t \quad \mathcal{N}(0,1)$. (A quick way of seeing this is to note that by rotational symmetry, we may assume WLOG that $w = e_1$, in which case the calculation is very straightforward.)

In what follows, we say $\widehat{\phi}_a$ are the Hermite coefficients of $\phi : \mathbb{R} \to \mathbb{R}$ if $\phi = \sum_a \widehat{\phi}_a H_a$, where the H_a are the normalized probabilists' Hermite polynomials. We use $\tilde{H}_a$ to denote the un-normalized (i.e. monic) Hermite polynomials. (Note that this is somewhat nonstandard notation.)

First we consider ReLUs.

Lemma C.3.1. $\widehat{\mathrm{ReLU}}_0 = 1/\sqrt{2\pi}$, $\widehat{\mathrm{ReLU}}_1 = 1/2$ *and for* $a \quad 2$, $\widehat{\mathrm{ReLU}}_a = \frac{1}{\sqrt{2\pi a!}}(\tilde{H}_a(0) + a\tilde{H}_{a-2}(0))$. *In particular,* $\widehat{\mathrm{ReLU}}_a = 0$ *for odd* $a \quad 3$ *and* $|\widehat{\mathrm{ReLU}}_a| = \Theta(a^{-5/4})$ *for even* a.

Proof. We use the following standard recurrence relation: $\tilde{H}_{a+1}(x) = x\tilde{H}_a(x)$

$a\tilde{H}_{a-1}(x)$. For $a \geq 2$,

$$\begin{aligned}
\widehat{\mathrm{ReLU}}_a &= \frac{1}{\sqrt{2\pi}} \int_{-1}^{1} \mathrm{ReLU}(x) H_a(x) e^{-\frac{x^2}{2}} dx \\
&= \frac{1}{\sqrt{2\pi a!}} \int_{0}^{1} x\tilde{H}_a(x) e^{-\frac{x^2}{2}} dx \\
&= \frac{1}{\sqrt{2\pi a!}} \int_{0}^{1} (\tilde{H}_{a+1}(x) + a\tilde{H}_{a-1}(x)) e^{-\frac{x^2}{2}} dx \\
&= \frac{1}{\sqrt{2\pi a!}} (\tilde{H}_a(0) + a\tilde{H}_{a-2}(0)).
\end{aligned}$$

Since $\tilde{H}_a(0) = 0$ for odd a, $\widehat{\mathrm{ReLU}}_a = 0$ as well. For even $a = 2b$ with $b \geq 2$, by standard expressions for $\tilde{H}_a(0)$, we have

$$\begin{aligned}
\widehat{\mathrm{ReLU}}_a &= \frac{1}{\sqrt{2\pi(2b)!}} (\tilde{H}_{2b}(0) + 2b\tilde{H}_{2b-2}(0)) \\
&= \frac{1}{\sqrt{2\pi(2b)!}} \left((-1)^b \frac{(2b)!}{b!2^b} + 2b(-1)^{b-1} \frac{(2b-2)!}{(b-1)!2^{b-1}} \right) \\
&= \frac{(-1)^b \sqrt{(2b)!}}{\sqrt{2\pi} b!2^b} \left(1 - \frac{2b}{2b-1} \right) \\
&= \frac{(-1)^{b+1} \sqrt{(2b)!}}{\sqrt{2\pi}(2b-1) b!2^b} \\
&\leq \frac{(-1)^{b+1}}{\sqrt{2\pi}(2b-1)(2b)^{1/4}} \\
&\leq \frac{(-1)^{b+1}}{b^{5/4}}
\end{aligned}$$

Here the second inequality follows from the fact $\binom{n}{n/2} \leq \frac{2^{n/2}}{\sqrt{n}}$. □

Corollary C.3.2. *The δ-approximate degree of* ReLU *under $N(0,1)$ is $O(\delta^{-4/3})$.*

Proof. Let p denote the the Hermite expansion of ReLU truncated at degree d. By

the fact that $|\widehat{\mathrm{ReLU}}_a| = \Theta(a^{-5/4})$ for even a (and 0 for odd a), we see that

$$\begin{aligned}\|p - \mathrm{ReLU}\|^2 &= \sum_{a>d} \widehat{\mathrm{ReLU}}_a^2 \\ &= \sum_{\substack{a>d \\ a \text{ even}}} \Theta(a^{-5/2}) \\ &= \Theta(d^{-3/2}).\end{aligned}$$

For this to be at most δ^2, we only need $d = O(\delta^{-4/3})$. □

Now we turn to sigmoids. Let σ denote the standard sigmoid, i.e. the logistic function $\sigma(t) = 1/(1+e^{-t})$.

Lemma C.3.3. *For all sufficiently large a, $\widehat{\sigma}_a = e^{-\Theta(a)}$.*

Proof. Upper bounds on the Hermite coefficients of sigmoidal funtions are known to follow from classic results in the complex analysis of Hermite series [Hil40, Boy84]. We refer to [PSG19, Corollary F.7.1], where this computation is done for $\tanh'(x) = 1 - \tanh^2(x)$. The calculation is very similar for σ (in fact, σ is just an affine shift of tanh). □

Corollary C.3.4. *The δ-approximate degree of σ under $N(0,1)$ is $\tilde{O}(\log^2 1/\delta)$.*

Proof. Let p denote the Hermite expansion of σ truncated at degree d. Observe that

$$\begin{aligned}\|\sigma - p\|^2 &= \sum_{a>d} \widehat{\sigma}_a^2 \\ &= \sum_{a>d} e^{-\Theta(a)} \\ &= \Theta(e^{-\Theta(d)}),\end{aligned}$$

which is at most δ^2 for $d = \tilde{O}(\log^2 1/\delta)$. □

C.4 Frank–Wolfe convergence guarantee

Here we provide a self-contained proof of Theorem 4.2.5, restated here. In fact, we generalize the analysis to handle any constant factor approximation to the optimum, meaning that in the Frank–Wolfe subproblem of Algorithm 1, we only require

$$\langle s, \nabla p(z_t)\rangle \leq \alpha \max_{s' \in Z'} \langle s', \nabla p(z_t)\rangle, \quad \langle z_{t+1}, \nabla p(z_t)\rangle \leq \frac{1}{2}\delta\gamma_t C_p \tag{C.4.1}$$

for some constant $\alpha \geq 1$. We closely follow [Jag13, Appendix A], noting the differences in our slightly more general setup (the standard setup has $Z' = Z$, and $\alpha = 1$).

Theorem C.4.1. *Let $Z' \subseteq Z$ be convex sets, and let $p : Z \to \mathbb{R}$ be a β-smoothly convex function. Let $C_p = \beta \operatorname{diam}(Z)^2$. Suppose that $z^{(0)} \in Z'$ achieves $\min_{z' \in Z'} p(z')$. For every t, the iterates of Algorithm 1 (modified to work with Eq. (C.4.1)) satisfy*

$$p(z_t) - p(z^*) \leq \frac{2C_p}{\alpha^2(t+2)}(1+\delta).$$

Proof. Define the duality gap function $g : Z \to \mathbb{R}$ as

$$g(z) = \max_{s \in Z'} \langle z - s, \nabla p(z)\rangle.$$

Notice that g takes in any $z \in Z$ but maximizes only over $s \in Z'$. By convexity of p over Z, we know that for all $z \in Z, s \in Z'$, $p(z) + \langle s - z, \nabla p(z)\rangle \geq p(s)$, meaning that $p(z) - p(s) \leq g(z)$. In particular, $p(z) - p(z^*) \leq g(z)$, so that $g(z)$ always provides an upper bound on the gap between $p(z)$ and $p(z^*)$ — this is weak duality.

Next we establish the following guarantee on the progress made in each step, which corresponds to Lemma 5 in Jaggi's proof.

Claim. *Let the t^{th} step be $z_{t+1} = z_t + \gamma(s - z_t)$, where $z_t, z_{t+1}, s \in Z$, $\gamma \in [0,1]$ is arbitrary, and s satisfies*

$$\langle s, \nabla p(z_t)\rangle \leq \alpha \max_{s' \in Z'} \langle s', \nabla p(z_t)\rangle, \quad \langle z_{t+1}, \nabla p(z_t)\rangle \leq \frac{1}{2}\delta\gamma C_p.$$

Then we have

$$p(z_{t+1}) - p(z_t) \leq \alpha\gamma g(z_t) + \frac{\gamma^2}{2}C_p(1+\delta).$$

To see this, first note that because p is β-smoothly convex,

$$p(z_{t+1}) = p(z_t + \gamma(s \quad z_t))$$
$$p(z_t) + \gamma hs \quad z_t, \; r\, p(z_t) i + \frac{\gamma^2}{2} C_p.$$

And from the way s 2 Z was picked, we have

$$hs \quad z_t, \quad r\, p(z_t) i \quad \alpha \max_{s^0 2 Z^0} hs^0 \quad z_t, \quad r\, p(z_t) i \quad \frac{1}{2}\delta\gamma C_p$$
$$= \alpha q(z_t) \quad \frac{1}{2}\delta\gamma C_p.$$

The claim now follows.

As a consequence of the claim, we can say

$$p(z_{t+1}) \quad p(z\ ) \quad p(z_t) \quad p(z\ ) \quad \gamma q(z_t) + \frac{\gamma^2}{2} C_p(1+\delta)$$
$$(1 \quad \alpha\gamma)(p(z_t) \quad p(z\ )) + \frac{\gamma^2}{2} C_p(1+\delta),$$

since $q(z_t) \quad p(z_t) \quad p(z\ )$ (weak duality). Taking $\gamma = \gamma_t = \frac{2}{\alpha(t+2)}$, the following bound can now by proven by induction on t:

$$p(z_t) \quad p(z\ ) \quad \frac{2}{\alpha^2(t+2)} C_p(1+\delta).$$

This proves the theorem. □

C.5 Relationship between Boolean 0-1 loss and real-valued correlation loss

Let D be a distribution on $\mathsf{R}^n \quad \mathsf{R}$. Our lower bound applies against agnostic learners that satisfy Assumption 4.3.1, with a real-valued correlation guarantee, i.e. learners that learn a class H by outputting $f : \mathsf{R}^n \; ! \; \mathsf{R}$ such that

$$\mathop{\mathsf{E}}_{(x,y) \; D}[f(x)y] \quad \max_{g 2 H} \mathop{\mathsf{E}}_{(x,y) \; D}[g(x)y] \quad \epsilon. \tag{C.5.1}$$

In the Boolean setting, where the labels are $\{-1,1\}$-valued, we have a distribution P on $\mathbb{R}^n \times \{-1,1\}$. A learner is said to agnostically learn H in terms of 0-1 loss if it is able to output $f : \mathbb{R}^n \to \{-1,1\}$ such that

$$\Pr_{(a,b)\sim P}[f(a) \neq b] \leq \min_{g\in H} \Pr_{(a,b)\sim P}[g(a) \neq b] + \epsilon,$$

or equivalently

$$\mathbb{E}_{(a,b)\sim P}[f(a)b] \geq \max_{g\in H} \mathbb{E}_{(a,b)\sim P}[g(a)b] - \epsilon/2,$$

since $\mathbb{E}_{(a,b)\sim P}[f(a)b] = 1 - 2\Pr_{(a,b)\sim P}[f(a) \neq b]$. (The latter formulation has the benefit of making sense even for real-valued $f : \mathbb{R}^n \to \mathbb{R}$.)

It is not obvious that a learner L of the above kind (with a Boolean 0-1 loss guarantee) gives us a real-valued correlation loss guarantee, because it only knows how to operate on distributions P on $\mathbb{R}^n \times \{-1,1\}$ (with Boolean labels), not distributions D on $\mathbb{R}^n \times \mathbb{R}$ (with arbitrary real labels). Moreover, in the SQ setting, we must be able to translate L's queries to P, which are of the form $\phi : \mathbb{R}^n \times \{-1,1\} \to \mathbb{R}$, into queries to D. We claim that both of these difficulties can be gotten around. We will show that if D has bounded labels, say in $[-C, C]$, we can construct a distribution P on $\mathbb{R}^n \times \{-1,1\}$ and simulate L on P to obtain a correlation loss guarantee wrt D.

Indeed, let D denote the marginal of D on $\mathbb{R}^n$; for us, D is always $N(0, I_n)$. Then P can be constructed simply as follows: draw $a \sim D$, and then randomly pick $b \in \{-1,1\}$ such that $\mathbb{E}[b|a] = (\mathbb{E}_{(x,y)\sim D}[y|x = a])/C$. (One could think of this as the "p-concept trick".) Equivalently, pick

$$b = \begin{cases} 1 & \text{with probability } \frac{1+(\mathbb{E}_{(x,y)\sim D}[y|x=a])/C}{2} \\ -1 & \text{otherwise} \end{cases}$$

One can easily see that for any $f : \mathbb{R}^n \to \mathbb{R}$,

$$\mathbb{E}_{(a,b)\sim P}[f(a)b] = \frac{1}{C}\mathbb{E}_{(x,y)\sim D}[f(x)y],$$

so that using L to learn up to 0-1 error ϵ gives a correlation loss guarantee up to $C\epsilon/2$. It remains to show that we can indeed simulate L's queries to P using only

SQ access to D. For any query $\phi : \mathbb{R}^n \to [-1, 1] \to \mathbb{R}$, observe that (since the marginal of P on $\mathbb{R}^n$ is also D)

$$\begin{aligned}\mathop{\mathbb{E}}_{(a,b)\sim P}[\phi(a,b)] &= \mathop{\mathbb{E}}_{a\sim D}\left[\phi(a,1)\frac{1+(\mathbb{E}_{(x,y)\sim D}[y|x=a])/C}{2} + \phi(a,-1)\frac{1-(\mathbb{E}_{(x,y)\sim D}[y|x=a])/C}{2}\right] \\ &= \frac{1}{2}\mathop{\mathbb{E}}_{a\sim D}[\phi(a,1)+\phi(a,-1)] + \frac{1}{2C}\mathop{\mathbb{E}}_{(x,y)\sim D}[(\phi(x,1)-\phi(x,-1))y].\end{aligned}$$

This expression can be computed using two statistical queries to D (or even just one, since we know the marginal D).

In our reduction (Theorem 4.4.1), we end up using the base learner on labeled distributions D where the labels correspond to the loss functional's gradient; when using surrogate loss, the label for x is $\psi(f(x)) - \psi(f(x))$. We see that this is indeed bounded in $[-2, 2]$, since $\psi : \mathbb{R} \to [-1, 1]$. Recall that in solving the Frank–Wolfe subproblem we needed to worry about simulating SQ access to this D using only SQ access to the true $D_{\psi \circ f}$ (see Eq. (4.4.1) and surrounding discussion). Here we actually have a further layer: we need to simulate SQ access to P using SQ access to D, itself simulated using actual SQ access to $D_{\psi \circ f}$. But it is easily verified that by the argument just outlined, no trouble arises here, and that one can in fact also "directly" simulate P using $D_{\psi \circ f}$ by the same argument as used for Eq. (4.4.1).

C.6 Relationship between square loss and correlation loss for ReLUs

Let D be a distribution on $\mathbb{R}^n \times \mathbb{R}$, and assume the labels are bounded in $[-C, C]$. Our lower bounds apply to agnostic learners that satisfy Assumption 4.3.1, with a guarantee in terms of correlation, where the output hypothesis f must satisfy

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[f(x)y] \geq \max_{g\in H} \mathop{\mathbb{E}}_{(x,y)\sim D}[g(x)y] - \epsilon.$$

But agnostic learning of real-valued functions is usually phrased in terms of square loss:

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[(f(x)-y)^2] \leq \min_{g\in H} \mathop{\mathbb{E}}_{(x,y)\sim D}[(g(x)-y)^2] + \epsilon^0.$$

Here we show that for the class of ReLUs, $H = H_{\mathsf{ReLU}}$, an agnostic learner L with a square loss guarantee can be used to satisfy Assumption 4.3.1. Fundamentally, this amounts to working out a geometric relationship between distances and projections in our function space, and much of the following argument can be viewed as a somewhat careful elaboration of what, in the familiar Euclidean setup, is more easily visualized.

For simplicity, throughout this section we will scale the class H_{ReLU} so that the maximum norm of any function is 1:

$$H = H_{\mathsf{ReLU}} = f \quad \frac{p}{2}\mathrm{ReLU}(u \quad x)\, j\, ku k_2 \quad 1g.$$

An important property of this class is that we can always scale a function $h \; 2 \; H$ to have any desired norm in $[0, 1]$ without leaving the class. That is, for any nonzero $h \; 2 \; H$ and any $\lambda \; 2 \; [0, 1]$, $\frac{\lambda}{khk} h \; 2 \; H$. This follows simply from the fact that $k\mathrm{ReLU}(u \quad x)k = ku k_2 / \frac{p}{2}$. We can think of this as saying that H is a norm-bounded section of a convex cone.

Let $f_{\mathsf{cmf}}(x) = \mathsf{E}[yjx]$. Let h_{sq} be a minimizer over all $h \; 2 \; H$ of the squared loss, $\mathsf{E}_{(x,y) \;\; D}[(h(x) \quad y)^2]$. An equivalent and more convenient view is that this is a minimizer of the squared distance $kh \quad f_{\mathsf{cmf}}k^2$, since

$$kh \quad f_{\mathsf{cmf}}k^2 = khk^2 \quad 2\hbar h, f_{\mathsf{cmf}}\, i + kf_{\mathsf{cmf}}k^2 = \underset{D}{\mathsf{E}}[(h(x) \quad y)^2] + kf_{\mathsf{cmf}}k^2 \quad \underset{D}{\mathsf{E}}[y^2],$$

and the latter terms are independent of h. This view is particularly important since it, combined with the fact that H is essentially a bounded convex cone, gives us an orthogonal projection theorem. Specifically, it is the case that the norm of h_{sq} must be the length of the projection of f_{cmf} onto the line λh_{sq} for $\lambda \; 2 \; [0, 1]$ (assuming this length is at most 1; otherwise, the norm is 1). In other words,

$$kh_{\mathsf{sq}}k = \min f\hbar \frac{h_{\mathsf{sq}}}{kh_{\mathsf{sq}}k}, f_{\mathsf{cmf}}\, i, 1g. \tag{C.6.1}$$

This can be seen by asking: for what $\lambda \; 2 \; [0, 1]$ is $\left\| \frac{\lambda}{kh_{\mathsf{sq}}k} h_{\mathsf{sq}} \quad f_{\mathsf{cmf}} \right\|$ minimized? (The point being that h_{sq} could be rescaled to have norm λ.) By writing this as

$$\left\| \frac{\lambda}{kh_{\mathsf{sq}}k} h_{\mathsf{sq}} \quad f_{\mathsf{cmf}} \right\|^2 = \left(\lambda \quad \hbar \frac{h_{\mathsf{sq}}}{kh_{\mathsf{sq}}k}, f_{\mathsf{cmf}}\, i \right)^2 + kf_{\mathsf{cmf}}k^2 \quad \hbar \frac{h_{\mathsf{sq}}}{kh_{\mathsf{sq}}k}, f_{\mathsf{cmf}}\, i^2,$$

the observation follows immediately.[1] This projection theorem also tells us that $h_{\mathsf{sq}} = 0$ iff f_{cmf} has no projection onto any $h \in H$, i.e. $\langle h, f_{\mathsf{cmf}} \rangle = 0$ for all $h \in H$.[2]

Let h_{cor} be a maximizer of the correlation, $\mathbb{E}_{(x,y)\sim D}[h(x)y] = \langle h, f_{\mathsf{cmf}} \rangle$. We may clearly assume that h_{cor} has the maximum possible norm, which is 1. We claim that in fact, h_{cor} can be taken to be $h_{\mathsf{sq}}/\|h_{\mathsf{sq}}\|$ (assuming $h_{\mathsf{sq}} \neq 0$; otherwise, $h_{\mathsf{cor}} = 0$ as well since, as noted, this means $\langle h, f_{\mathsf{cmf}} \rangle = 0$ for all $h \in H$). To see why, first assume $h_{\mathsf{sq}} \neq 0$ and use the fact that for any nonzero $h \in H$, the square loss achieved by $\frac{\|h_{\mathsf{sq}}\|}{\|h\|} h$ (i.e. h scaled to have h_{sq}'s norm) cannot be better than that of h_{sq} itself. Thus by an algebraic manipulation we have

$$\begin{aligned} & \|h_{\mathsf{sq}} - f_{\mathsf{cmf}}\|^2 \quad \|\frac{\|h_{\mathsf{sq}}\|}{\|h\|} h - f_{\mathsf{cmf}}\|^2 \\ \Leftrightarrow \quad & \langle \frac{h_{\mathsf{sq}}}{\|h_{\mathsf{sq}}\|}, f_{\mathsf{cmf}} \rangle \quad \langle \frac{h}{\|h\|}, f_{\mathsf{cmf}} \rangle \quad \langle h, f_{\mathsf{cmf}} \rangle. \end{aligned}$$

Since this holds for any $h \in H$, we may take $h_{\mathsf{cor}} = h_{\mathsf{sq}}/\|h_{\mathsf{sq}}\|$.

Now suppose we have an agnostic learner in terms of square loss that returns h such that

$$\|h - f_{\mathsf{cmf}}\|^2 \quad \|h_{\mathsf{sq}} - f_{\mathsf{cmf}}\|^2 + \epsilon'.$$

For a suitable choice of ϵ' (depending on the final desired ϵ), we would like to say that $h/\|h\|$ achieves correlation that is ϵ-competitive with h_{cor}. Indeed, if $h_{\mathsf{sq}} = 0$ this is trivial, since as noted this means $\langle h, f_{\mathsf{cmf}} \rangle = 0$ for all $h \in H$. Otherwise, by comparing $\frac{\|h\|}{\|h_{\mathsf{sq}}\|} h_{\mathsf{sq}}$ (i.e. h_{sq} scaled to have h's norm) with h_{sq} itself, we may say that

$$\|h - f_{\mathsf{cmf}}\|^2 \quad \|h_{\mathsf{sq}} - f_{\mathsf{cmf}}\|^2 + \epsilon' \quad \|\frac{\|h\|}{\|h_{\mathsf{sq}}\|} h_{\mathsf{sq}} - f_{\mathsf{cmf}}\|^2 + \epsilon'.$$

[1] Note that here we are assuming $\langle h_{\mathsf{sq}}, f_{\mathsf{cmf}} \rangle \quad 0$ WLOG, since otherwise we would consider $-h_{\mathsf{sq}}$.

[2] For another way to see this, for any nonzero $h \in H$, expand $\|\lambda h - f_{\mathsf{cmf}}\|^2 \quad \|0 - f_{\mathsf{cmf}}\|^2$ and let $\lambda \to 0$.

Some rearrangement gives

$$\begin{aligned} \left\| \frac{h}{\|h\|}, f_{\mathrm{cmf}} \right\| \quad \left\| \frac{h_{\mathrm{sq}}}{\|h_{\mathrm{sq}}\|}, f_{\mathrm{cmf}} \right\| \quad \frac{\epsilon^{\emptyset}}{2\|h\|} \\ = \left\| h_{\mathrm{cor}}, f_{\mathrm{cmf}} \right\| \quad \frac{\epsilon^{\emptyset}}{2\|h\|}, \end{aligned} \tag{C.6.2}$$

showing that $h/\|h\|$ is $\frac{\epsilon^{\emptyset}}{2\|h\|}$-competitive with h_{cor}.

But an issue here is that $\|h\|$ could be very small, or even zero. We claim that we can actually address this separately as an easy case: it implies that we are in a trivial situation in which even the 0 function performs fairly well, and so even the best possible correlation must be quite small.

Lemma C.6.1. *Let h be such that $\|h\| \leq \|f_{\mathrm{cmf}}\|^2 + \|h_{\mathrm{sq}} - f_{\mathrm{cmf}}\|^2 + \epsilon^{\emptyset}$. Suppose $\|h\| \geq \eta$. Then $\left\| h_{\mathrm{cor}}, f_{\mathrm{cmf}} \right\| \geq \sqrt{\epsilon^{\emptyset} + 2C\eta}$. In particular, the 0 function is $\sqrt{\epsilon^{\emptyset} + 2C\eta}$-competitive with h_{cor}.*

Proof. By Cauchy–Schwarz,

$$\left\| 0 - f_{\mathrm{cmf}} \right\|^2 \leq \left\| h - f_{\mathrm{cmf}} \right\|^2 = \left\| h, f_{\mathrm{cmf}} \right\| \leq \|f_{\mathrm{cmf}}\|^2 \leq 2\|h\| \|f_{\mathrm{cmf}}\|^2 \leq 2C\eta,$$

where we use $\|f_{\mathrm{cmf}}\| \leq C$ since the labels are assumed to be bounded in $[-C, C]$. Thus

$$\|0 - f_{\mathrm{cmf}}\|^2 \leq \|h - f_{\mathrm{cmf}}\|^2 + 2C\eta \leq \|h_{\mathrm{sq}} - f_{\mathrm{cmf}}\|^2 + \epsilon^{\emptyset} + 2C\eta.$$

On the other hand, by definition of h_{sq},

$$\|h_{\mathrm{sq}} - f_{\mathrm{cmf}}\|^2 \leq \|0 - f_{\mathrm{cmf}}\|^2,$$

Put together, this means that the 0 function achieves nearly the same square loss as h_{sq}:

$$\|h_{\mathrm{sq}} - f_{\mathrm{cmf}}\|^2 \leq \|0 - f_{\mathrm{cmf}}\|^2 \leq \|h_{\mathrm{sq}} - f_{\mathrm{cmf}}\|^2 + \epsilon^{\emptyset} + 2C\eta. \tag{C.6.3}$$

This lets us conclude that $\|h_{\mathrm{sq}}\|$ must be small:

$$\|h_{\mathrm{sq}}\|^2 = \|f_{\mathrm{cmf}}\|^2 \leq \|h_{\mathrm{sq}}\| \|f_{\mathrm{cmf}}\|^2 + 2\|h_{\mathrm{sq}}\| \left\| f_{\mathrm{cmf}}, h_{\mathrm{sq}} \right\| \leq \epsilon^{\emptyset} + 2C\eta,$$

where we use Eq. (C.6.3) and the fact that by can rewrite Eq. (C.6.1) as $kh_{sq}k$ $h\frac{h_{sq}}{kh_{sq}k}, f_{cmf}i$, or hh_{sq} $f_{cmf}, h_{sq}i$ 0. But now since $kh_{sq}k$ $\sqrt{\epsilon^0 + 2C\eta} < 1$ (ϵ^0 and η will be picked sufficiently small), Eq. (C.6.1) boils down to saying that

$$hh_{cor}, f_{cmf}i = h\frac{h_{sq}}{kh_{sq}k}, f_{cmf}i = kh_{sq}k \quad \sqrt{\epsilon^0 + 2C\eta}.$$

□

We can now put everything together.

Theorem C.6.2. *Suppose we have an agnostic learner L for H_{ReLU} under D with a square loss guarantee. Then L can be used to yield a correlation guarantee, i.e. to satisfy Assumption 4.3.1.*

Proof. Run L with $\epsilon^0 = \Theta(\epsilon^3)$ to get h such that kh $f_{cmf}k^2$ kh_{sq} $f_{cmf}k^2 + \epsilon^0$. By Lemma C.6.1, if khk $\eta = \Theta(\epsilon^2)$, then 0 is ϵ-competitive with h_{cor}. So we may assume that khk $\Theta(\epsilon^2)$. But then by Eq. (C.6.2), since now $\frac{\epsilon^0}{2khk}$ ϵ, we get that h/khk is ϵ-competitive with h_{cor}. □

Appendix D: A Moment-Matching Approach to Testable Learning and a New Characterization of Rademacher Complexity

D.1 Proof of strong duality in Theorem 6.3.2

We will use the following statement of conic duality, specialized to the setting of moment problems.

Theorem D.1.1 ([Sha01, Section 3]). *Let $\Omega = \mathbb{R}^d$, endowed with the standard Borel sigma algebra, and let C be the set of all nonnegative Borel measures on Ω. Pair the space of signed measures on Ω and functions mapping Ω to $\mathbb{R}$ using the following inner product: $\langle g, \mu \rangle = \int g \, d\mu$. Let $\phi, \psi_1, \ldots, \psi_p : \Omega \to \mathbb{R}$ be functions, let $b \in \mathbb{R}^p$, let $A : \mu \mapsto (\langle \psi_1, \mu \rangle, \ldots, \langle \psi_p, \mu \rangle)$, and let K be a closed convex cone in $\mathbb{R}^p$.*

Define the following primal problem ([Sha01, Eq 3.2]):

$$\sup_{\mu \in C} \langle \phi, \mu \rangle \quad \text{subject to} \quad A\mu - b \in K. \tag{D.1.1}$$

Let $K^* = \{\alpha \mid \alpha^\top \beta \geq 0 \ \forall \beta \in K\}$ be the polar cone of K. Then the dual is defined as follows ([Sha01, Eq 3.8]):

$$\inf_{\alpha \in K^*} b^\top \alpha \quad \text{subject to} \quad \sum_{i=1}^{p} \alpha_i \psi_i(\omega) - \phi(\omega) \ \forall \omega \in \Omega. \tag{D.1.2}$$

Further, a sufficient condition for strong duality to hold (i.e. for both primal and dual to have the same finite optimum) is that b lie in the interior of the feasible set, i.e. $b \in \{\widetilde{b} \mid \exists \mu \in C : A\mu - \widetilde{b} \in K\}$ ([Sha01, Eq 3.12]).

Let Ω, C and the dual pairing $\langle \cdot, \cdot \rangle$ be as above. Note that C can also be viewed as the convex cone generated by all Dirac measures on Ω, and also that when μ is a probability measure (i.e., nonnegative and with total measure 1), $\langle g, \mu \rangle = \mathbb{E}_\mu[g]$.

Our goal now is to obtain strong duality between Eqs. (6.3.1) and (6.3.3) as a consequence of Theorem D.1.1. Let $r = |I(k,d)| - 1$, and for convenience write $I(k,d) = \{I_0, I_1, \ldots, I_r\}$, where $I_0 = (0, \ldots, 0)$. Define the functions $\psi_1, \ldots, \psi_r : \Omega \to \mathbb{R}$ to be the nontrivial monomials corresponding to $I(k,d)$, i.e., $\psi_j(x) = x_{I_j}$, and define $\psi_{r+j} = -\psi_r$ for all $1 \le j \le r$. Let $K \subseteq \mathbb{R}^{2r+1}$ be the following convex cone: $K = \{0\} \times \mathbb{R}^{2r}$, where $\mathbb{R}_- = (-\infty, 0]$. Let A be a linear operator on $\mathcal{C}$ given by

$$A\mu = (\langle 1, \mu\rangle, \langle \psi_1, \mu\rangle, \ldots, \langle \psi_r, \mu\rangle, \langle -\psi_1, \mu\rangle, \ldots, \langle -\psi_r, \mu\rangle) \tag{D.1.3}$$

and let $b \in \mathbb{R}^{2r+1}$ be given by

$$b = (1, \sigma_{I_1} + \Delta_{I_1}, \ldots, \sigma_{I_r} + \Delta_{I_r}, \quad -\sigma_{I_1} + \Delta_{I_1}, \ldots, \quad -\sigma_{I_r} + \Delta_{I_r}). \tag{D.1.4}$$

We claim that our original primal LP Eq. (6.3.1) corresponds to the following conic linear program, which has the form of Eq. (D.1.1), with $p = 2r + 1$ and $\phi = f$:

$$\sup_{\mu \in \mathcal{C}} \langle f, \mu\rangle \quad \text{subject to} \quad A\mu \le b \in K \tag{D.1.5}$$

Indeed, the first coordinate of b ensures the $I = 0$ constraint, namely that $\langle 1, \mu\rangle = 1$ and hence μ is a valid probability measure (note that the cone $\mathcal{C}$ already only consists of nonnegative measures), and the other coordinates ensure that $\sigma_I - \Delta_I \le \mathbb{E}_\mu[x_I] \le \sigma_I + \Delta_I$ for every other $I \in I(k,d) \setminus \{0\}$.

The dual of this program may be written in the form of Eq. (D.1.2) as follows. First introduce dual variables $\alpha_0 \in \mathbb{R}$ (corresponding to the first constraint), and $(\alpha_1, \ldots, \alpha_{2r}) \in \mathbb{R}^{2r}$ (corresponding to the others), and write $b = (b_0, \ldots, b_{2r})$. The dual is

$$\inf_{\alpha \in K^*} \alpha_0 b_0 + \sum_{j=1}^{2r} \alpha_j b_j \quad \text{subject to} \quad \alpha_0 + \sum_{j=1}^{2r} \alpha_j \psi_j \ge f \text{ over } \Omega. \tag{D.1.6}$$

Here K^* is the polar cone of K, and is easily seen to be $K^* = \mathbb{R} \times \mathbb{R}^{2r}$. This means $K^* = \mathbb{R} \times \mathbb{R}^{2r}_+$, i.e. $\alpha_0 \in \mathbb{R}$ and $(\alpha_1, \ldots, \alpha_{2r}) \in \mathbb{R}^{2r}_+$. The dual objective may be

simplified as follows:

$$\alpha_0 b_0 + \sum_{j=1}^{2r} \alpha_j b_j = \alpha_0 + \sum_{j=1}^{r} \big(\alpha_j(\sigma_{I_j} + \Delta_{I_j}) + \alpha_{r+j}(\quad \sigma_{I_j} + \Delta_{I_j})\big) \tag{D.1.7}$$

$$= \alpha_0 + \sum_{j=1}^{r} (\alpha_j \quad \alpha_{r+j})\sigma_{I_j} + \sum_{j=1}^{r} (\alpha_j + \alpha_{r+j})\Delta_{I_j}. \tag{D.1.8}$$

The constraint simplifies to

$$\alpha_0 + \sum_{j=1}^{r} (\alpha_j \quad \alpha_{j+r})\psi_j \quad f.$$

To simplify this further, if we let $\beta_j = \alpha_j \quad \alpha_{j+r}$ for every $1 \quad j \quad r$, then it is not hard to see that the objective is minimized when each $\alpha_j + \alpha_{j+r} = \jmath\beta_j\jmath$ (in particular, when $\alpha_j = \max\{\beta_j, 0\}$ and $\alpha_{r+j} = \max\{\quad \beta_j, 0\}$). Thus if we also let $\beta_0 = \alpha_0$, then the dual objective becomes $\beta_0 + \sum_{r=1}^{j} \beta_j\sigma_{I_j} + \sum_{j=1}^{r} \jmath\beta_j\jmath\Delta_{I_j}$, and the constraint becomes $\beta_0 + \sum_{j=1}^{r} \beta_j\psi_j \quad f$. Recalling that $\sigma_{I_0} = 1$ and $\Delta_{I_0} = \Delta_0 = 0$, this is precisely the dual we originally claimed, Eq. (6.3.3).

Now, by Theorem D.1.1, a sufficient condition for strong duality is that b lie in the interior of the feasible set, i.e. $b \in \{\widetilde{b} \mid \exists \mu \in C : A\mu \quad \widetilde{b} \in K\}$. This means that for any sufficiently small perturbation $\widetilde{b}$ of b, there must exist a measure $\mu \in C$ such that $A\mu \quad \widetilde{b} \in K$, i.e. with $\widetilde{b}$ as its approximate vector of moments up to order k. We argue this slightly informally as follows. Let μ denote D from the statement of Theorem 6.3.2. Suppose

$$\widetilde{b} = b + \eta = (b_0 + \eta_0, b_1 + \eta_1, \ldots, b_{r+1} + \eta_{r+1}, \ldots) \tag{D.1.9}$$

$$= (1 + \eta_0, \sigma_{I_1} + \Delta_{I_1} + \eta_1, \ldots, \quad \sigma_{I_1} + \Delta_{I_1} + \eta_{r+1}, \ldots), \tag{D.1.10}$$

where $\eta_0, \ldots, \eta_{2r} \in \mathbb{R}^{2r+1}$ are to be thought of as small. The condition that $A\mu \quad \widetilde{b} \in K$ is the same as saying that μ satisfies the following:

$$\langle\mu, 1\rangle = 1 + \eta_0 \tag{D.1.11}$$

$$\sigma_{I_j} \quad \Delta_{I_j} \quad \eta_{r+j} \quad \langle\mu, \psi_j\rangle \quad \sigma_{I_j} + \Delta_{I_j} + \eta_j \quad \forall 1 \quad j \quad r. \tag{D.1.12}$$

For sufficiently small η, we claim that a small perturbation of μ will continue to satisfy these conditions. First, note that because $\langle \mu, 1 \rangle \neq 1$, μ is no longer formally a probability measure. But for sufficiently small η_0, by adding or removing some mass to μ arbitrarily close to the origin, we can increase or decrease its total mass while keeping all its moments nearly unchanged (because the ψ_j are continuous and $\psi_j(0) = 0$ for all $j \neq 0$, and the new mass is essentially all at 0). Take μ to be such a perturbation of μ, satisfying $\langle \mu, 1 \rangle = 1 + \eta_0$. We have just argued that for every $j \neq 0$, $\langle \mu, \psi_j \rangle$ differs from $\langle \mu, \psi_j \rangle = \sigma_{I_j}$ by an arbitrarily small amount. Thus if $\eta_1, \ldots, \eta_{2r}$ are sufficiently small (it suffices to have each $\eta_j < \Delta_{I_j}/2$), then the approximate moment matching conditions will still be satisfied by μ, because there is still a slack of at least $\Delta_{I_j}/2 > 0$ in the constraint arising from I_j. This establishes that b is indeed in the interior of the feasible set, and hence that strong duality holds between Eq. (6.3.1) and Eq. (6.3.3).

Bibliography

[AAK21] Naman Agarwal, Pranjal Awasthi, and Satyen Kale. A deep conditioning treatment of neural networks. In *Algorithmic Learning Theory*, pages 249–305. PMLR, 2021.

[ADHV19] Alexandr Andoni, Rishabh Dudeja, Daniel Hsu, and Kiran Vodrahalli. Attribute-efficient learning of monomials over highly-correlated variables. In *Thirtieth International Conference on Algorithmic Learning Theory*, 2019.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003.

[AK95] Dana Angluin and Michael Kharitonov. When won't membership queries help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited. In *Annual Cryptology Conference*, pages 57–74. Springer, 2013.

[APVZ14] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning sparse polynomial functions. In *Proceedings of the twenty- fth annual ACM-SIAM symposium on Discrete algorithms*, pages 500–510. SIAM, 2014.

[AS20] Emmanuel Abbe and Colin Sandon. Poly-time universality and limitations of deep learning. *arXiv preprint arXiv:2001.02992*, 2020.

[ATV21] Pranjal Awasthi, Alex Tang, and Aravindan Vijayaraghavan. Efficient algorithms for learning depth-2 neural networks with general relu activations. *Advances in Neural Information Processing Systems*, 34, 2021.

[Bac17] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[Bar14] Peter L Bartlett. UC Berkeley CS281B/Stat241B: Statistical Learning Theory, Lecture 7, 2014. URL: https://www.stat.berkeley.edu/~bartlett/courses/2014spring-cs281bstat241b/lectures/07-notes.pdf. Last visited on 2022/10/29.

[Baz09] Louay MJ Bazzi. Polylogarithmic independence can fool dnf formulas. *SIAM Journal on Computing*, 38(6):2220–2272, 2009.

[BB20] Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

[BBH+20] Matthew Brennan, Guy Bresler, Samuel B Hopkins, Jerry Li, and Tselil Schramm. Statistical query algorithms and low-degree tests are almost equivalent. *arXiv preprint arXiv:2009.06107*, 2020.

[BBL02] Peter L Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48(1):85–113, 2002.

[BBL03] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer school on machine learning*, pages 169–207. Springer, 2003.

[BBM05] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[BDES02] Shai Ben-David, Nadav Eiron, and Hans Ulrich Simon. Limitations of learning via embeddings in euclidean half spaces. *Journal of Machine Learning Research*, 3(Nov):441–461, 2002.

[BDH+20] Ainesh Bakshi, Ilias Diakonikolas, Samuel B. Hopkins, Daniel Kane, Sushrut Karmalkar, and Pravesh K. Kothari. Outlier-robust clustering of gaussians and other non-spherical mixtures. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 149–159. IEEE, 2020.

[BdW01] Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings 16th Annual IEEE Conference on Computational Complexity*, pages 120–130. IEEE, 2001.

[BDW02] Harry Buhrman and Ronald De Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[BE85] D. Bakry and Michel Émery. Diffusions hypercontractives. In *Seminaire de probabilites, XIX, 1983/84*, volume 1123 of *Lecture Notes in Math.*, pages 177–206. Springer, Berlin, 1985.

[Bel21] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

[BF02] Nader H Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2(Feb):359–395, 2002.

[BFJ+94] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

[BG17a] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *CoRR*, abs/1702.07966, 2017.

[BG17b] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 605–614, 2017.

[BGM+16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography Conference*, pages 209–224. Springer, 2016.

[BGML+18] Sauvik Bhattacharya, Oscar Garcia-Morchon, Thijs Laarhoven, Ronald Rietman, Markku-Juhani O Saarinen, Ludo Tolhuizen, and Zhenfei Zhang. Round5: Compact and fast post-quantum public-key encryption. *IACR Cryptol. ePrint Arch.*, 2018:725, 2018.

[BHKL15] Alina Beygelzimer, Elad Hazan, Satyen Kale, and Haipeng Luo. Online gradient boosting. In *Advances in neural information processing systems*, pages 2458–2466, 2015.

[BI91] Gyora M Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theoretical Computer Science*, 86(2):377–389, 1991.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J Wu. Exploring crypto dark matter. In *Theory of Cryptography Conference*, pages 699–729. Springer, 2018.

[BJW19] Ainesh Bakshi, Rajesh Jayaram, and David P Woodruff. Learning two layer rectified neural networks in polynomial time. In *Conference on Learning Theory*, pages 195–268. PMLR, 2019.

[BK20] Ainesh Bakshi and Pravesh Kothari. Outlier-robust clustering of non-spherical mixtures. *CoRR*, abs/2005.02970, 2020.

[BK21] Ainesh Bakshi and Pravesh K. Kothari. List-decodable subspace recovery: Dimension independent error in polynomial time. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1279–1297. SIAM, 2021.

[BL97] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Arti cial intelligence*, 97(1-2):245–271, 1997.

[BLLT20] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[BLMT23] Guy Blanc, Jane Lange, Ali Malik, and Li-Yang Tan. Lifting uniform learners via distributional decomposition. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1755–1767, 2023.

[Blu90] Avrim Blum. Learning boolean functions in an infinite attribute space. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 64–72, 1990.

[BM97] Lucien Birgé and Pascal Massart. From model selection to adaptive estimation. In *Festschrift for lucien le cam*, pages 55–87. Springer, 1997.

[BM02] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[BMR21] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

[Bog21] Andrej Bogdanov. Personal communication, 2021.

[Boy84] John P Boyd. Asymptotic coefficients of hermite function series. *Journal of Computational Physics*, 54(3):382–410, 1984.

[BP14] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In *Annual Cryptology Conference*, pages 353–370. Springer, 2014.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 719–737. Springer, 2012.

[BR89] Avrim Blum and Ronald L Rivest. Training a 3-node neural network is NP-complete. In *Advances in neural information processing systems*, pages 494–501, 1989.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer, 2017.

[Bra10] Mark Braverman. Polylogarithmic independence fools AC^0 circuits. *Journal of the ACM (JACM)*, 57(5):1–10, 2010.

[BRST21] Joan Bruna, Oded Regev, Min Jae Song, and Yi Tang. Continuous lwe. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 694–707, 2021.

[BT96] Nader H Bshouty and Christino Tamon. On the fourier spectrum of monotone functions. *Journal of the ACM (JACM)*, 43(4):747–770, 1996.

[BT19] Mark Bun and Justin Thaler. A Nearly Optimal Lower Bound on the Approximate Degree of AC^0. *SIAM Journal on Computing*, 49(4):FOCS17–59–FOCS17–96, 2019.

[BW08] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(8), 2008.

[CDM16] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 67–82. Springer, 2016.

[CGMK22] Sitan Chen, Aravind Gollakota, Raghu Meka, and Adam Klivans. Hardness of noise-free learning for two-hidden-layer neural networks. *Advances in Neural Information Processing Systems*, 2022.

[CGV15] Aloni Cohen, Shafi Goldwasser, and Vinod Vaikuntanathan. Aggregate pseudorandom functions and connections to learning. In *Theory of Cryptography Conference*, pages 61–89. Springer, 2015.

[CKLS18] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. Lizard: Cut off the tail! a practical post-quantum public-key encryption from lwe and lwr. In *International Conference on Security and Cryptography for Networks*, pages 160–177. Springer, 2018.

[CKM20] Sitan Chen, Adam R Klivans, and Raghu Meka. Learning deep relu networks is fixed-parameter tractable. *arXiv preprint arXiv:2009.13512*, 2020.

[CKM21] Sitan Chen, Adam Klivans, and Raghu Meka. Efficiently learning one hidden layer relu networks from queries. In *Advances in Neural Information Processing Systems*, 2021.

[CST+00] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[Dan16a] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 105–117, 2016.

[Dan16b] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 105–117, 2016.

[DFGK17] Carlos M Da Fonseca, M Lawrence Glasser, and Victor Kowalenko. Basic trigonometric power sums with applications. *The Ramanujan Journal*, 42(2):401–428, 2017.

[DG21] Amit Daniely and Elad Granot. An exact poly-time membership-queries algorithm for extraction a three-layer relu network. *arXiv preprint arXiv:2105.09673*, 2021.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010.

[DGK+20] Ilias Diakonikolas, Surbhi Goel, Sushrut Karmalkar, Adam Klivans, and Mahdi Soltanolkotabi. Approximation Schemes for ReLU Regression. In *Conference on Learning Theory*, 2020. To appear.

[DGKP20] Abhimanyu Das, Sreenivas Gollapudi, Ravi Kumar, and Rina Panigrahy. On the learnability of random deep networks. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 398–410. SIAM, 2020.

[DK20] Ilias Diakonikolas and Daniel M. Kane. Small covers for near-zero sets of polynomials and learning latent variable models. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–195, 2020.

[DK21] Ilias Diakonikolas and Daniel M. Kane. Non-gaussian component analysis via lattice basis reduction, 2021.

[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.

[DKKZ20] Ilias Diakonikolas, Daniel Kane, Vasilis Kontonis, and Nikos Zarifis. Algorithms and SQ Lower Bounds for PAC Learning One-Hidden-Layer ReLU Networks. In *Conference on Learning Theory*, 2020. To appear.

[DKN10] Ilias Diakonikolas, Daniel M Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2010.

[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. In *Conference on Learning Theory*, pages 1552–1584. PMLR, 2021.

[DKR23] Ilias Diakonikolas, Daniel Kane, and Lisheng Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals. In *International Conference on Machine Learning*, pages 7922–7938. PMLR, 2023.

[DKRV18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure

encryption and cca-secure kem. In *International Conference on Cryptology in Africa*, pages 282–305. Springer, 2018.

[DKS17] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017.

[DKZ20a] 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*, 33:13586–13596, 2020.

[DKZ20b] Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-Optimal SQ Lower Bounds for Agnostically Learning Halfspaces and ReLUs under Gaussian Marginals. In *Advances in Neural Information Processing Systems*, 2020.

[DLSS14] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 441–448, 2014.

[DNS22] Anindya De, Shivam Nadimpalli, and Rocco Servedio. Convex influences. *Innovations in Theoretical Computer Science*, 2022.

[DS16] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *Conference on Learning Theory*, pages 815–830, 2016.

[DSFT+14] Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of the twenty-sixth annual*

ACM-SIAM symposium on Discrete algorithms, pages 498–511. SIAM, 2014.

[DSS16] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *Conference on Learning Theory*, pages 815–830. PMLR, 2016.

[DV20a] Amit Daniely and Gal Vardi. Hardness of learning neural networks with natural weights. *arXiv preprint arXiv:2006.03177*, 2020.

[DV20b] Amit Daniely and Gal Vardi. Hardness of learning neural networks with natural weights. *Advances in Neural Information Processing Systems*, 33, 2020.

[DV21] Amit Daniely and Gal Vardi. From local pseudorandom generators to hardness of learning. In *Conference on Learning Theory*, pages 1358–1394. PMLR, 2021.

[Ear19] Mike Earnest. Proving an identity involving the alternating sum of products of binomial coefficients. Mathematics Stack Exchange, 2019. URL: https://math.stackexchange.com/q/3108805 (version: 2019-02-11).

[FCG20] Spencer Frei, Yuan Cao, and Quanquan Gu. Agnostic learning of a single neuron with gradient descent. *arXiv preprint arXiv:2005.14426*, 2020.

[Fel08] Vitaly Feldman. Evolvability from learning algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 619–628, 2008.

[Fel09] Vitaly Feldman. On the power of membership queries in agnostic learning. *The Journal of Machine Learning Research*, 10:163–182, 2009.

[Fel10] Vitaly Feldman. Distribution-specific agnostic boosting. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 241–250. Tsinghua University Press, 2010.

[Fel12] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.

[Fel14] Vitaly Feldman. Open problem: The statistical query complexity of learning sparse halfspaces. In *Conference on Learning Theory*, pages 1283–1289, 2014.

[Fel16] Vitaly Feldman. Statistical query learning. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 2090–2095. Springer New York, New York, NY, 2016.

[Fel17] Vitaly Feldman. A general characterization of the statistical query complexity. In *Conference on Learning Theory*, pages 785–830, 2017.

[FGR+17] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):8, 2017.

[FGRW12] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

[Fri01] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[Gab81] Yu R Gabovich. Stability of the characterization of the multivariate normal distribution in the skitovich-darmois theorem. *Journal of Soviet Mathematics*, 16(5):1341–1349, 1981.

[GGJ+20] Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial Lower Bounds for Learning One-Layer Neural Networks using Gradient Descent. In *International Conference on Machine Learning*, 2020. To appear.

[GGK20] Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. *Advances in Neural Information Processing Systems*, 33:2147–2158, 2020.

[GKK19] Surbhi Goel, Sushrut Karmalkar, and Adam Klivans. Time/accuracy tradeoffs for learning a relu with respect to gaussian marginals. In *Advances in Neural Information Processing Systems*, pages 8582–8591, 2019.

[GKK23] Aravind Gollakota, Adam R Klivans, and Pravesh K Kothari. A moment-matching approach to testable learning and a new characterization of rademacher complexity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 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.

[GKKT17a] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In *Conference on Learning Theory*, pages 1004–1042, 2017.

[GKKT17b] Surbhi Goel, Varun Kanade, Adam R. Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In *COLT*, pages 1004–1042, 2017.

[GKM18] Surbhi Goel, Adam R. Klivans, and Raghu Meka. Learning one convolutional layer with overlapping patches. In *ICML*, volume 80, pages 1778–1786. PMLR, 2018.

[GKSV23a] Aravind Gollakota, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. An efficient tester-learner for halfspaces. *arXiv preprint arXiv:2302.14853*, 2023.

[GKSV23b] Aravind Gollakota, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Tester-learners for halfspaces: Universal algorithms. *arXiv preprint arXiv:2305.11765*, 2023.

[GLM18a] Rong Ge, Jason D. Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. In *ICLR*. OpenReview.net, 2018.

[GLM18b] Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[GR09] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

[GRSY21] Shafi Goldwasser, Guy N Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

[Hil40] Einar Hille. Contributions to the theory of hermitian series ii. the representation problem. *Transactions of the American Mathematical Society*, 47(1):80–94, 1940.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

[HMRT22] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

[HS07] Lisa Hellerstein and Rocco A Servedio. On pac learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007.

[HS19] Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to AC^0. *Random Structures & Algorithms*, 54(2):289–303, 2019.

[Hua19] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3):949–955, 2019.

[IK22] Misha Ivkov and Pravesh K. Kothari. List-decodable covariance estimation. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1276–1283. ACM, 2022.

[Jac97] Jeffrey C. Jackson. An efficient membership-query algorithm for learning dnf with respect to the uniform distribution. *J. Comput. Syst. Sci*, 55(3):414–440, 1997.

[Jag13] Martin Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

[JCB+20] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 1345–1362. USENIX Association, 2020.

[JHG18] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NeurIPS*, pages 8580–8589, 2018.

[JSA15] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

[JWZ20] Rajesh Jayaram, David P. Woodruff, and Qiuyi Zhang. Span recovery for deep neural networks with applications to input obfuscation. In *ICLR*. OpenReview.net, 2020.

[JZ16] Zhengzhong Jin and Yunlei Zhao. Optimal key consensus in presence of noise. *arXiv preprint arXiv:1611.06150*, 2016.

[Kan11] Daniel M Kane. The gaussian surface area and noise sensitivity of degree-d polynomial threshold functions. *computational complexity*, 20(2):389–412, 2011.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[Kha93] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the twenty- fth annual ACM symposium on Theory of computing*, pages 372–381, 1993.

[Kha95] Michael Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *Journal of Computer and System Sciences*, 50(3):600–610, 1995.

[KK09] Varun Kanade and Adam Kalai. Potential-based agnostic boosting. In *Advances in neural information processing systems*, pages 880–888, 2009.

[KK14a] Adam Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2014.

[KK14b] Adam R. Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *APPROX-RANDOM*, volume 28 of *LIPIcs*, pages 793–809. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[KK21] Adam Tauman Kalai and Varun Kanade. Efficient learning with arbitrary covariate shift. In *Algorithmic Learning Theory*, pages 850–864. PMLR, 2021.

[KKK19] Sushrut Karmalkar, Adam R. Klivans, and Pravesh Kothari. List-decodable linear regression. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *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.

[KKM13] Daniel Kane, Adam Klivans, and Raghu Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. In *Conference on Learning Theory*, pages 522–545. PMLR, 2013.

[KKMS08] Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[KL01] Matthias Krause and Stefan Lucks. Pseudorandom functions in in tc0 and cryptographic limitations to proving lower bounds. *computational complexity*, 10(4):297–313, 2001.

[KL18] Pravesh K Kothari and Roi Livni. Improper learning by refuting. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[KLS95] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13:541–559, 1995.

[KM13] Adam Klivans and Raghu Meka. Moment-matching polynomials. *arXiv preprint arXiv:1301.0820*, 2013.

[KMS20] Pritish Kamath, Omar Montasser, and Nathan Srebro. Approximate is good enough: Probabilistic variants of dimensional and margin complexity. In *Conference on Learning Theory*, 2020.

[Kol01] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[Kol06] Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

[KOS08] Adam R Klivans, Ryan O'Donnell, and Rocco A Servedio. Learning geometric concepts via gaussian surface area. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 541–550. IEEE, 2008.

[KP00] Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

[KR96] LB Klebanov and ST Rachev. Proximity of probability measures with common marginals in a finite number of directions. *Lecture Notes-Monograph Series*, pages 162–174, 1996.

[KR21] Gil Kur and Alexander Rakhlin. On the minimal error of empirical risk minimization. In *Conference on Learning Theory*, pages 2849–2852. PMLR, 2021.

[KS94a] Michael J Kearns and Robert E Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[KS94b] Michael J Kearns and Robert E Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[KS04] Adam R Klivans and Rocco A Servedio. Learning DNF in time $2^{O(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004.

[KS07] Adam R Klivans and Alexander A Sherstov. Unconditional lower bounds for learning intersections of halfspaces. *Machine Learning*, 69(2-3):97–114, 2007.

[KS09] Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

[KS10] Adam R Klivans and Alexander A Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010.

[KSP20] Rohan Karthikeyan, Siddharth Sinha, and Vallabh Patil. On the resolution of the sensitivity conjecture. *Bulletin of the American Mathematical Society*, 2020.

[KSS92] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. In *Proceedings of the fth annual workshop on Computational learning theory*, pages 341–352, 1992.

[KSS94] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[KV94a] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

[KV94b] Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[Led01] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2001.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261:515–534, 1982.

[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.

[LMSS07] Nati Linial, Shahar Mendelson, Gideon Schechtman, and Adi Shraibman. Complexity measures of sign matrices. *Combinatorica*, 27(4):439–463, 2007.

[LMZ20] Yuanzhi Li, Tengyu Ma, and Hongyang R. Zhang. Learning over-parametrized two-layer neural networks beyond ntk. In *Conference on Learning Theory 2020*, volume 125, pages 2613–2682. PMLR, 2020.

[LSSS14] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

[LY17] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems 30*, pages 597–607, 2017.

[MBBF00] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.

[MOS03] Elchanan Mossel, Ryan O'Donnell, and Rocco P Servedio. Learning juntas. In *Proceedings of the thirty- fth annual ACM symposium on Theory of computing*, pages 206–212, 2003.

[MP69] M.L. Minsky and S. Papert. *Perceptrons; an Introduction to Computational Geometry*. MIT Press, 1969.

[MS22] Eran Malach and Shai Shalev-Shwartz. When hardness of approximation meets hardness of learning. *The Journal of Machine Learning Research*, 23(1):3942–3965, 2022.

[MSDH19] Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *FAT*, pages 1–9. ACM, 2019.

[NK19] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 458–467. IEEE, 1997.

[NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational complexity*, 4(4):301–313, 1994.

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 468–474, 1992.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, mar 2016.

[PMG+17] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM, 2017.

[PSG19] Abhishek Panigrahi, Abhishek Shetty, and Navin Goyal. Effect of activation functions on the training of overparametrized neural nets. *arXiv preprint arXiv:1908.05660*, 2019.

[PSP17] PSPACEhard. Alternating sum of binomial coefficients identity. Mathematics Stack Exchange, 2017. URL: https://math.stackexchange.com/q/2183223 (version: 2017-03-12).

[Raz92] Alexander A Razborov. On small depth threshold circuits. In *Scandinavian Workshop on Algorithm Theory*, pages 42–52. Springer, 1992.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7(30):11, 2010.

[Rey20] Lev Reyzin. Statistical queries and statistical algorithms: Foundations and applications. *arXiv preprint arXiv:2004.00557*, 2020.

[RK20] David Rolnick and Konrad P. Kording. Reverse-engineering deep relu networks. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8178–8187. PMLR, 2020.

[RKSF13] Svetlozar T Rachev, Lev B Klebanov, Stoyan V Stoyanov, and Frank Fabozzi. *The methods of distances in the theory of probability and statistics*, volume 10. Springer, 2013.

[Rob55] Herbert Robbins. A remark on stirling's formula. *The American mathematical monthly*, 62(1):26–29, 1955.

[RR97] Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[RS10] Alexander A Razborov and Alexander A Sherstov. The sign-rank of ac^0. *SIAM Journal on Computing*, 39(5):1833–1855, 2010.

[RV22] Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. *Proceedings of the fty- fth annual ACM Symposium on Theory of Computing*, 2022. To appear.

[RV23] Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1643–1656, 2023.

[RY20a] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 161–180. SIAM, 2020.

[RY20b] Prasad Raghavendra and Morris Yau. List decodable subspace recovery, 2020.

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[SF12] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[Sha01] Alexander Shapiro. On duality theory of conic linear problems. In *Semi-in nite programming*, pages 135–165. Springer, 2001.

[Sha18a] Ohad Shamir. Distribution-specific hardness of learning neural networks. *J. Mach. Learn. Res*, 19:32:1–32:29, 2018.

[Sha18b] Ohad Shamir. Distribution-specific hardness of learning neural networks. *The Journal of Machine Learning Research*, 19(1):1135–1163, 2018.

[She08a] Alexander A. Sherstov. Communication lower bounds using dual polynomials. *Bulletin of the EATCS*, 2008.

[She08b] Alexander A Sherstov. Halfspace matrices. *Computational Complexity*, 17:149–178, 2008.

[She11] Alexander A Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011.

[She12] Alexander A Sherstov. Making polynomials robust to noise. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 747–758, 2012.

[She13] Alexander A Sherstov. Optimal bounds for sign-representing the intersection of two halfspaces by polynomials. *Combinatorica*, 33(1):73–96, 2013.

[SSSS10] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

[SSSS17] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3067–3075, 2017.

[ST17] Rocco A Servedio and Li-Yang Tan. What circuit classes can be learned with non-trivial savings? In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[STT12] Rocco Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning andweight-degree tradeoffs for polynomial threshold functions. In *Conference on Learning Theory*, pages 14–1, 2012.

[SVWX17] Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. In *Advances in Neural Information Processing Systems*, pages 5514–5522, 2017.

[SW14] Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: a review. *Statistics surveys*, 8:45, 2014.

[SW19] Alexander A Sherstov and Pei Wu. Near-optimal lower bounds on the threshold degree and sign-rank of ac0. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 401–412, 2019.

[SZB21] Min Jae Song, Ilias Zadik, and Joan Bruna. On the cryptographic hardness of learning single periodic neurons. *arXiv preprint arXiv:2106.10744*, 2021.

[Szö09] Balázs Szörényi. Characterizing statistical query learning: simplified notions and proofs. In *International Conference on Algorithmic Learning Theory*, pages 186–200. Springer, 2009.

[Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of AC^0. In *32nd Computational Complexity Conference (CCC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[Tia17] Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70, pages 3404–3413. PMLR, 2017.

[Tie23] Stefan Tiegel. Hardness of agnostically learning halfspaces from worst-case lattice problems. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 3029–3064. PMLR, 2023.

[TJ+16] Florian Tramèr, Fan Zhang 0022, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. *CoRR*, abs/1609.02943, 2016.

[Tsy08] A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, 2008.

[Vad17] Salil Vadhan. On learning vs. refutation. In *Conference on Learning Theory*, pages 1835–1848. PMLR, 2017.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vap98] V.N. Vapnik. *Statistical Learning Theory*. A Wiley-Interscience publication. Wiley, 1998.

[Vap00] Vladimir N. Vapnik. *The nature of statistical learning theory*. Statistics for Engineering and Information Science. Springer-Verlag, New York, second edition, 2000.

[VdG00] Sara A Van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[VRPS21] Gal Vardi, Daniel Reichman, Toniann Pitassi, and Ohad Shamir. Size and depth separation in approximating natural functions with neural networks. *arXiv preprint arXiv:2102.00314*, 2021.

[VSS+22] Kiran Vodrahalli, Rakesh Shivanna, Mahesh Sathiamoorthy, Sagar Jain, and Ed Chi. Algorithms for efficiently learning low-rank neural networks, 2022.

[Vu98] Van H Vu. On the infeasibility of training neural networks with small mean-squared error. *IEEE Transactions on Information Theory*, 44(7):2892–2900, 1998.

[Vu06] VH Vu. On the infeasibility of training neural networks with small mean-squared error. *IEEE Transactions on Information Theory*, 44(7):2892–2900, 2006.

[VW19a] Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *COLT*, volume 99, 2019.

[VW19b] Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pages 3115–3117, 2019.

[YS19] Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2019.

[YS20] Gilad Yehudai and Ohad Shamir. Learning a single neuron with gradient methods. *arXiv preprint arXiv:2001.05205*, 2020.

[ZBH+21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[Zol84] Vladimir Mikhailovich Zolotarev. Probability metrics. *Theory of Probability & Its Applications*, 28(2):278–302, 1984.

[ZPS17] Qiuyi Zhang, Rina Panigrahy, and Sushant Sachdeva. Electron-proton dynamics in deep learning. *CoRR*, abs/1702.00458, 2017.

[ZSJ+17a] Kai Zhong, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *ICML*, volume 70, pages 4140–4149. JMLR.org, 2017.

[ZSJ+17b] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR, 2017.

[ZSWB22] Ilias Zadik, Min Jae Song, Alexander S. Wein, and Joan Bruna. Lattice-based methods surpass sum-of-squares in clustering, 2022.

[ZYWG19a] Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Arti cial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1524–1534. PMLR, 2019.

[ZYWG19b] Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. In *The 22nd International Conference on Arti cial Intelligence and Statistics*, pages 1524–1534. PMLR, 2019.