Learning Deep ReLU Networks Is Fixed-Parameter Tractable

Sitan Chen
UC Berkeley
sitanc@berkeley.edu

Adam R. Klivans
UT Austin
klivans@cs.utexas.edu

Raghu Meka
UCLA
raghum@cs.ucla.edu

2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-2055-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS52979.2021.00073

***Abstract*—We consider the problem of learning an unknown ReLU network with respect to Gaussian inputs and obtain the first nontrivial results for networks of depth more than two. We give an algorithm whose running time is a fixed polynomial in the ambient dimension and some (exponentially large) function of only the network's parameters. Our results provably cannot be obtained using gradient-based methods and give the first example of a class of efficiently learnable neural networks that gradient descent will fail to learn.**

Our bounds depend on the number of hidden units, depth, spectral norm of the weight matrices, and Lipschitz constant of the overall network (we show that some dependence on the Lipschitz constant is necessary). We also give a bound that is doubly exponential in the size of the network but is independent of spectral norm.

In contrast, prior work for learning networks of depth three or higher requires *exponential* time in the ambient dimension, even when the above parameters are bounded by a constant. Additionally, all prior work for the depth-two case requires well-conditioned weights and/or positive coefficients to obtain efficient run-times. Our algorithm does not require these assumptions.

Our main technical tool is a type of filtered PCA that can be used to iteratively recover an approximate basis for the subspace spanned by the hidden units in the first layer. Our analysis leverages new structural results on lattice polynomials from tropical geometry.

***Keywords*-deep learning; supervised learning; multiple index models; regression; statistical query; gradient descent**

I. Introduction

We study the problem of learning the following class of concepts:

Definition I.1 (ReLU Networks)**.** Let $\mathcal{C}_S$ denote the concept class of (feedforward) ReLU networks over $\mathbb{R}^d$ of size S. Specifically, $F \in \mathcal{C}_S$ if there exist weight matrices $\mathbf{W}_0 \in \mathbb{R}^{k_0 \times d}, \mathbf{W}_1 \in \mathbb{R}^{k_1 \times k_0}, \ldots, \mathbf{W}_L \in \mathbb{R}^{k_L \times k_{L-1}}, \mathbf{W}_{L+1} \in \mathbb{R}^{1 \times k_L}$ for which

$$F(x) \triangleq \mathbf{W}_{L+1}\phi\left(\mathbf{W}_L\phi\left(\cdots\phi(\mathbf{W}_0 x)\cdots\right)\right),$$

where $\phi(z) \triangleq \max(z, 0)$ is the ReLU activation applied entrywise, and $k_0 + \cdots + k_L = S$. In this case we say that F is computed by a ReLU network with depth $L + 2$. We will refer to the rank of $\mathbf{W}_0$ as k, to emphasize that the value of F only depends on a k-dimensional subspace of $\mathbb{R}^d$. We will also let $k_{L+1} = 1$.

When the weight matrices of two ReLU networks $F, F' \in \mathcal{C}_S$ have the same dimensions (at all layers), then we say that F and F' *have the same architecture*.

For example, a depth two ReLU network of size S in d-dimensions is a function $F : \mathbb{R}^d \to \mathbb{R}$ of the form

$$F(x) = \sum_{i=1}^{S} \lambda_i \phi(\langle w_i, x\rangle),$$

where $\lambda_i \in \mathbb{R}$ are scalars and $w_i \in \mathbb{R}^d$ are arbitrary vectors.

Note that any Boolean function $F : \{\pm 1\}^n \to \{\pm 1\}$ can be computed by an n-layer ReLU network. In particular, if F is a junta depending only on k variables, then it can be computed by a k-layer ReLU network with size that depends only on k.

Learning ReLU Networks: The problem of PAC learning an unknown ReLU network from labeled examples is a central challenge in the theory of machine learning. Given samples from a distribution of the form $(x, y) \in \mathbb{R}^d \times \mathbb{R}$ where $y = F(x)$ with F an unknown size-S ReLU network,[1] and x is drawn according to a distribution $\mathcal{D}$, the goal is to output a function $f : \mathbb{R}^d \to \mathbb{R}$ with small *test* error, i.e., $\mathbb{E}_{x,y}[(y - f(x))^2] \leq \varepsilon\, \mathbb{E}[y^2]$. In this work, we focus on the widely studied case where the input distribution on x is Gaussian.

Ideally, we would like an algorithm with sample complexity and running time that is polynomial in all the relevant parameters. Even for learning arbitrary sums of ReLUs, i.e. depth two ReLU networks where we additionally assume the $\mathbf{W}_1$ has all positive entries, it remains a major open question to obtain a polynomial-time algorithm (see [1] for the strongest-known result). As a first step, one could ask for an algorithm that at least depends polynomially on the *ambient dimension* (it is often easy to obtain brute-force search algorithms that run in time exponential in the dimension[2]). In the absence of additional assumptions however, even this goal has remained elusive: it was not known how to achieve a subexponential-time algorithm even for learning *general* depth two ReLU networks, let alone ReLU networks of higher depth.

[1]It should not be difficult to extend our techniques to the setting where $y = F(x) + \mathcal{N}(0, \sigma^2)$, but we focus on the noiseless case for simplicity in this work.

[2]Although in our specific case even this type of search turns out to be nontrivial.

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00073

In this work, we address this gap by giving the first algorithm for learning ReLU networks whose running time is a fixed polynomial in the dimension, regardless of the depth of the network. Our algorithm is *fixed-parameter tractable*: we show that we can *properly learn* (i.e., the output hypothesis is also a ReLU network) ReLU networks with sample complexity and running time that is a fixed polynomial in the dimension and an exponential function of the network's *parameters*.

More precisely, our main result is as follows. We will also make the (as it turns out necessary) assumption that the ReLU network has a bounded Lipschitz constant: a function $f : \mathbb{R}^d \to \mathbb{R}$ is Λ-Lipschitz if $|f(x) - f(x')| \leq \Lambda \|x - x'\|_2$ for all x, x'.

Theorem I.2 (Main, informal)**.** *Let $\mathcal{D}$ be the distribution over pairs $(x, y) \in \mathbb{R}^d \times \mathbb{R}$ where $x \sim \mathcal{N}(0, \mathrm{Id})$ and $y = F(x)$ for a size-S ReLU network F with depth $L + 2$, Lipschitz constant at most Λ, rank of* bottom *weight matrix $\mathbf{W}_0$ being k, and whose weight matrices all have spectral norm at most B.*

There is an algorithm that draws $d \log(1/\delta) \exp\left(\mathrm{poly}(k, S, \Lambda/\varepsilon)\right) B^{O(Lk)}$ samples, runs in time $\widetilde{O}(d^2 \log(1/\delta)) \exp\left(\mathrm{poly}(k, S, \Lambda/\varepsilon)\right) B^{O(LkS^2)}$, and outputs a ReLU network $\widetilde{F}$ such that $\mathbb{E}[(y - \widetilde{F}(x))^2] \leq \varepsilon$ with probability at least $1 - \delta$.[3]

Note that the sample complexity is linear while the runtime is quadratic in the ambient dimension. In particular, in the well-studied special case where the product of the spectral norms of the weight matrices is a constant (see e.g. [2]), in which case the Lipschitz constant of the network is also constant, we can obtain the following result as an immediate consequence of the formal version of the above theorem:

Corollary I.3. *Let $\mathcal{D}$ be the distribution over pairs $(x, y) \in \mathbb{R}^d \times \mathbb{R}$ where $x \sim \mathcal{N}(0, \mathrm{Id})$ and $y = F(x)$ for a size-S ReLU network F for which the product of the spectral norms of its weight matrices is a constant.*

Then there is an algorithm that draws $N = d \log(1/\delta) \exp(O(k^3/\varepsilon^2 + kS))$ samples, runs in time $\widetilde{O}(d^2 \log(1/\delta)) \exp(O(k^3 S^2/\varepsilon^2 + kS^3))$, and outputs a ReLU network $\widetilde{F}$ such that $\mathbb{E}[(y - \widetilde{F}(x))^2] \leq \varepsilon$ with probability at least $1 - \delta$.

As mentioned earlier, no algorithms that were sub-exponential in d were known even for S, B, ε being constants.

Before going further, we note that a dependence on the Lipschitz constant of the network is necessary even for learning depth two ReLU networks with respect to Gaussians:

[3]See full version for a discussion of why this guarantee is scale-invariant.

Example I.4. *Let $\Lambda > 0$. Consider the size-3, depth two ReLU network $F : \mathbb{R}^2 \to \mathbb{R}$ given by*

$$F(x_1, x_2) = \phi(x_1 + \Lambda x_2) + \phi(3x_1 + \Lambda x_2) - 2\phi(-x_1 + \Lambda x_2).$$

The Lipschitz constant of F is $\Theta(\Lambda)$: $F(0, 1/\Lambda) = 1$ and $F(1, 1/\Lambda) = 2$. Furthermore, note that for $(x_1, x_2) \in \mathbb{S}^1$, $F(x_1, x_2) = 0$ unless $x_2 \in [-3/\Lambda, 3/\Lambda]$. By rotational symmetry, for $(x_1, x_2) \sim \mathcal{N}(0, \mathrm{Id})$, $F(x_1, x_2) \neq 0$ with probability at most $O(1/\Lambda)$.

Note that for depth two ReLU networks with positive weights, no such dependence on the Lipschitz constant is necessary intuitively because without cancellations between the hidden units, one cannot devise "spiky" functions F which simultaneously have small variance but attain a large value at some bounded-norm x.

Interestingly, our techniques are also general enough to handle the more general family of all continuous piecewise-linear functions (see Definition IV.2 for a formal definition):

Theorem I.5. *Let $\mathcal{D}$ be the distribution over pairs $(x, y) \in \mathbb{R}^d \times \mathbb{R}$ where $x \sim \mathcal{N}(0, \mathrm{Id})$ and $y = F(x)$ for a continuous piecewise-linear function F which only depends on the projection of x to a k-dimensional subspace V, has at most M linear pieces, and is Λ-Lipschitz.*

There is an algorithm that draws $d \log(1/\delta) \cdot \mathrm{poly}\left(\exp\left(k^3 \Lambda^2/\varepsilon^2\right), M^k\right)$ samples, runs in time $\widetilde{O}(d^2 \log(1/\delta)) \cdot M^{M^2} \cdot \mathrm{poly}\left(\exp\left(k^4 \Lambda^2/\varepsilon^2\right), M^{k^2}\right)$, and outputs a piecewise-linear function $\widetilde{F}$ such that $\mathbb{E}[(y - \widetilde{F}(x))^2] \leq \varepsilon$ with probability at least $1 - \delta$.

Note that a size-S ReLU network is a continuous piecewise-linear function with at most 2^S linear pieces. Specializing Theorem I.5 to ReLU networks gives a guarantee which is incomparable to Theorem I.2: we obtain an algorithm that depends doubly exponentially on S but has no dependence on the norms of the weight matrices.

A. Prior Work on Provably Learning Neural Networks

Algorithmic Results: Algorithms for learning neural networks (obtaining small *test error*) have been intensely studied in the literature. In the last few years alone there have been many papers giving provable results for learning restricted classes of neural networks under various settings [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [1].

The predominant techniques are spectral or tensor-based dimension reduction [3], [5], [16], [24], kernel methods [4], [7], [25], [15], [17], and gradient-based methods [13], [14], [19]. All prior work takes distributional and/or architectural assumptions, the most common one being that the inputs come from a standard Gaussian. We will also work in this

setting.[4]

As pointed out in [26], [21], all existing algorithmic results for Gaussian inputs hold *only for depth two networks* and make at least one of two assumptions on the unknown network F in question:

- Weight matrix $\mathbf{W}_0$ is well-conditioned and, in particular, full rank.
- The vector at the output layer ($\mathbf{W}_1$ when $L = 0$) has all positive entries.

Assumption (1) allows one to use tensor decomposition to recover the parameters of the network and hence PAC learn, an idea that has inspired a long line of works [3], [5], [13], [14], [16]. However, the assumption is not necessary for PAC learning or achieving low-prediction error. For instance, consider a pathological case where $\mathbf{W}_0$ has repeated rows. Here, while parameter recovery is not possible it is still possible to PAC learn. To our knowledge, the only work that can PAC learn depth two networks over Gaussian inputs without a condition number bound on $\mathbf{W}_0$ is [24]. However, their work still requires assumption (2) (and only holds for depth two networks). Our work shows that assumption (2) is neither information-theoretically nor computationally necessary.

Limitations of Gradient-Based Methods: Two recent works [26], [24] showed that a broad family of algorithms, namely *correlational statistical query (CSQ) algorithms*, fail to PAC learn even depth two ReLU networks; that is, functions of the form $F(x) = \sum_{i=1}^{k} \lambda_i \phi(\langle v_i, x\rangle)$ with respect to Gaussian inputs in time polynomial in d where d is the ambient dimension (in fact, [24] rules out running time $d^{o(k)}$). Informally, a CSQ algorithm is limited to using noisy estimates of statistics of the form $\mathbb{E}[y \cdot \sigma(x)]$ for arbitrary bounded σ, where the expectation is over examples (x, y) and $y = F(x)$ is computed by the network. The point is that this already rules out a wide range of algorithmic approaches in theory and practice, including gradient descent on overparameterized networks (i.e., using neural tangent kernels [27] or the mean-field approximation for gradient dynamics [28]). Note that the algorithms of [24] for learning depth two ReLU networks with positive coefficients are CSQ algorithms as well.

Note that as a consequence of Theorem I.2, for any ε a function of k, our algorithm can learn the lower bound instances in [26], [24] to error ε in time $g(k) \cdot \mathrm{poly}(d)$ for some g (note that the norm bounds and Lipschitz constants for these instances are upper bounded by functions of k), which is impossible for any CSQ algorithm. We explain why our algorithm is not a CSQ algorithm in Section II.

For the classification version of this problem (i.e., taking a softmax) where we observe $Y \in \{0, 1\}$ such that $\mathbb{E}[Y|X] = \sigma(f(X))$ where σ is say sigmoid and $f(X)$ is a depth two ReLU network, Goel et al. [26] show that even general SQ algorithms cannot achieve a runtime with polynomial dependence on the dimension. We also remark there is an extensive literature of previous work showing various hardness results for learning certain classes of neural networks [29], [30], [31], [32], [7], [33], [34], [35], [19], [36], [37]. We refer the reader to [26] for a discussion of how these prior works relate to the above CSQ lower bounds.

[4]Other works such as [18] or kernel-based methods [4], [7] require strong norm-based assumptions on the inputs and weights.

B. Other Related Work and Discussion

Multi-Index Models: Functions computed by ReLU networks where $\mathbf{W}_0$ has fewer rows than columns are a special case of a *multi-index model*, that is, a function $F : \mathbb{R}^d \to \mathbb{R}$ given by $F(x) = f(\mathbf{W}^\top x)$ for some matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$ and some function $f : \mathbb{R}^k \to \mathbb{R}$. In the theoretical computer science literature, these are sometimes referred to as *subspace juntas* [38], [39], [40].

One result in this line of work which is close in spirit to the setting we consider is that of [41], which gives various conditions on f under which one can recover $\mathbf{W}$ (under Gaussian inputs) in the special case where $k = 1$, as well as a vector in the row span of $\mathbf{W}$ in the case of general k (although these results do not hold for ReLU). In general, the literature on multi-index models is vast, and we refer to [41] for a comprehensive overview of this body of work. Many works were inspired by a simple but powerful connection to Stein's lemma [42], [43], [44], which was also a key ingredient in the above algorithms for learning neural networks using tensor decomposition.

Another relevant line of work in this literature is the series of results on learning intersections of halfspaces (and indicators of convex sets more generally) over structured input distributions, see e.g. [45], [46], [47], [48], [49]. For Gaussian inputs, when the number of halfspaces (or more generally the dimension of the convex set's hidden subspace) is bounded, it was shown in [48] that one can essentially read off the row span of $\mathbf{W}$ from the eigendecomposition of $\mathbb{E}[y \cdot (xx^\top - \mathrm{Id})]$ where for a given x, $y = 0$ if x lies in the convex set and $y = 1$ otherwise. By the CSQ lower bounds of [26], [24], such an algorithm provably cannot learn general ReLU networks.

Alternatively, one could also try generalizing the approach of [48] to our real-valued setting by restricting to level sets S of the ReLU network and forming the matrix $\mathbb{E}[\mathbb{1}[x \in S](xx^\top - \mathrm{Id})]$. We remark however that the analysis in [48] for such an approach crucially uses convexity of the underlying concept and is therefore not applicable to our setting. Note that this technique is also known as *sliced inverse regression* [50], [51] in the multi-index model literature, and while it is related to the techniques that we employ, we explain in Remark II.1 why the state of the art here also falls short.

Non-Gaussian Component Analysis: As we discuss in Section II, the general approach we take is to find

careful reweightings of the distribution over x that will look non-Gaussian in some important direction, i.e., in the row span of $\mathbf{W}_0$. There have been several works on *non-Gaussian component analysis* (see, e.g., [52], [53] and the references therein), but this line of work is not relevant to our result. We also remark that the work [38] gives some moment-based conditions under which it is possible to learn multi-index models over Gaussian inputs via non-Gaussian component analysis. However, it seems highly nontrivial to verify whether such conditions hold for ReLU networks, and in addition, their results seem tailored to $\{0,1\}$-valued functions.

Piecewise-Linear Regression: We mention that previous works on *segmented regression* (see, e.g., [54] on the references therein) study regression for piecewise-linear functions but work with a different notion of piecewise-linearity that is unrelated to our setting.

Non-Homogeneous ReLU Networks: We leave as an open question whether our result can be extended to *non-homogeneous* networks of the form $F(x) \triangleq \mathbf{W}_{L+1}\phi(\mathbf{W}_L\phi(\cdots\phi(\mathbf{W}_0x + b_0) + b_1)\cdots + b_L)$, where $b_0,\ldots,b_L \in \mathbb{R}$ are unknown bias parameters. We stress that, over Gaussian inputs, we are not aware of any positive results even for learning non-homogeneous networks of depth two. As for negative results, the recent work of [55] rules out polynomial-time algorithms for learning non-homogeneous ReLU networks, even of depth three, assuming local PRGs with polynomial stretch and constant distinguishing advantage exist [56]. While this hardness result does not preclude the existence of a fixed-parameter tractable algorithm for non-homogeneous ReLU networks, it does give a compelling explanation for the lack of algorithmic progress in the non-homogeneous case.

II. Proof Overview

The conceptual novelty of our work is that we go beyond standard CSQ-based algorithms like gradient descent on square loss to give a fundamentally new algorithm for learning neural networks. There are a number of technical novelties to our approach we will describe over the course of outlining our algorithm and analysis in this section.

Suppose we are given samples (x,y) where $y = F(x)$ is computed by a size S ReLU network as in Definition I.1. Let $V \subseteq \mathbb{R}^d$ denote the span of the rows of $\mathbf{W}_0$ and let k be its dimension. We will call V the *relevant subspace*, because the value of F only depends on the projection of x to V. In particular, we can write $y = F'(\Pi_V(x))$ for some function $F' : V \to \mathbb{R}$ that is itself a size S ReLU network and Π_V denotes the projection operator onto V. The main focus of our algorithm will be in figuring out the relevant subspace V given samples (x,y). This is the hardest part of the algorithm, because once we learn the relevant subspace to high enough accuracy, we can grid-search over ReLU networks in this subspace. Even this grid search turns out to be non-trivial to analyze and entails proving new *stability* results for piecewise-linear functions.

Filtered PCA: Our algorithm builds upon the *filtered PCA* approach, originally introduced in [57] for the purposes of learning low-degree polynomials over Gaussian space.[5] For any $\psi : \mathbb{R} \to \mathbb{R}$, let $\mathbf{M}_\psi \triangleq \mathbb{E}[\psi(Y)(XX^T - \mathrm{Id})]$. A basic but important observation is that for any choice of ψ, all vectors orthogonal to the true subspace V are in the kernel of $\mathbf{M}_\psi$. A natural idea for identifying the true subspace then is to look at the nonzero singular vectors of $\mathbf{M}_\psi$ for a suitable ψ. If we could show that $\mathbf{M}_\psi$ has k nonzero singular values all bounded away from 0 by some dimension-independent margin $c(\psi)$, then we could hope to approximately recover V by empirically estimating $\mathbf{M}_\psi$ using $O(d/c(\psi)^2)$, invoking standard matrix concentration, and computing its top-k singular subspace. So the main hurdle is to identify an appropriate ψ for which this is the case.

What should the ψ be? For instance if ψ is the identity function, then the matrix $\mathbf{M}_\psi$ could be identically zero. This is an essential difference between our setting and the setting studied in previous works [24], [13] (in the $L = 0$ case) where the output layer's coefficients are all positive, for which this choice of ψ would suffice to recover the relevant subspace.

Note that this is consistent with the CSQ lower bounds of [26], [24], as any algorithm that just tries to use the spectrum of $\mathbf{M}_\psi$ for ψ being the identity function would be a CSQ algorithm. Indeed, for any of the 'hard' functions F from those works which are ReLU networks with $L = 0$ we would have $\mathbf{M}_\psi = 0$ if ψ is the identity function.

We will choose ψ not equal to the identity, and in this way our algorithm will be non-CSQ and evade the aforementioned CSQ lower bounds.

Threshold Filter.: Motivated by [57], our starting point in the present work is to consider ψ given by a univariate threshold, that is, $\psi(z) = \mathbb{1}[|z| > \tau]$ for suitable τ. For brevity, for $\tau \in \mathbb{R}$ define $\mathbf{M}_\tau = \mathbb{E}_{x,y}[\mathbb{1}[|y| > \tau](xx^T - \mathrm{Id})]$. Then we have that

$$\langle \Pi_V, \mathbf{M}_\tau \rangle = \mathop{\mathbb{E}}_{x,y}\big[\mathbb{1}[|y| > \tau] \cdot (\|\Pi_V x\|^2 - k)\big].$$

In particular, if one could choose τ for which $|F(x)| > \tau$ only if $\|\Pi_V x\|^2 \geq 2k$ [6], then we would conclude that $\langle \Pi_V, \mathbf{M}_\tau \rangle \geq k \cdot \mathbb{P}[|y| > \tau]$, so some singular value of $\mathbf{M}_\tau$ is at least $\mathbb{P}[|y| > \tau]$. If F is Λ-Lipschitz, we can simply choose τ to be $\sqrt{2k}\cdot\Lambda$, and provided $\mathbb{P}[|y| > \tau]$ is reasonably large, then we conclude that $\mathbf{M}_\tau$ has some reasonably large singular value. Finally, to lower bound $\mathbb{P}[|y| > \tau]$, we prove

[5] For readers familiar with the approach there, we explain in the full version why a straightforward application of the algorithm there cannot work, necessitating a far more involved approach in the present work.

[6] The choice of $2k$ here is for exposition; any bound noticeably more than k, e.g., $k+1$ will do.

an *anti-concentration* result for piecewise linear functions over Gaussian space (Lemma V.1).

In other words, if one conditions on the samples (x, y) whose responses y are sufficiently large in magnitude, then we show that the resulting distribution is noticeably non-Gaussian in some direction, and by taking the top singular vector of the conditional covariance, we can approximately recover some direction inside the relevant subspace V.

Unfortunately, all that the above analysis tells us is that the trace of $\mathbf{M}_\tau$ is non-negligible which in turn helps us guarantee that we identify at least one direction in V. It is not at all clear whether the above threshold approach is enough to identify more than just one vector in the relevant subspace. Indeed, recovering the full relevant subspace turns out to be significantly more challenging, and the core technical contribution of this work is to show how to do this.

Remark II.1 (Relation to Sliced Inverse Regression). The trick of conditioning only on (x, y) for which $|y|$ is sufficiently large is reminiscent of the technique of *slicing* originally introduced by [51] in the context of learning multi-index models. The high-level idea of slicing is that for any fixed value of y, the conditional law of $x|F(x) = y$ is likely to be non-Gaussian in most directions $v \in V$, so in particular, $\mathbb{E}[xx^\top - \mathrm{Id} \mid F(x) = y]$ should be nonzero, and its singular vectors will lie in V. This can be thought of as filtered PCA with the choice of function $\psi(z) = \mathbb{1}[z = y]$. The first issue with using such an approach to get an actual learning algorithm is that $\mathbb{P}_x[F(x) = y] = 0$ for any y, and the workaround in non-asymptotic analyses of sliced inverse regression [50] is to estimate something like $\mathbb{E}_y[\mathbb{E}[xx^\top - \mathrm{Id} \mid F(x) = y]]$ instead. While finite sample estimators for such objects are known, the conditions under which this approach can provably recover the relevant subspace are quite strong and not applicable to our setting.

Learning the Full Subspace: What Doesn't Work: One might hope that a more refined analysis shows that for a suitable τ, the spectrum of $\mathbf{M}_\tau$ can identify the entire subspace V. Given that we can already learn some $w \in V$ with the threshold approach above, a first step would be to try to find a direction in V orthogonal to w, by lower bounding the contribution to the Frobenius norm of $\mathbf{M}_\tau$ from vectors orthogonal to w. Concretely, letting $\Pi_{V\setminus\{w\}}$ denote the projector to the orthogonal complement of w in V, we have that $\langle \Pi_{V\setminus\{w\}}, \mathbf{M}_\tau\rangle = \mathbb{E}_{x,y}\big[\mathbb{1}[|y| > \tau] \cdot (\|\Pi_{V\setminus\{w\}}x\|^2 - (k-1))\big]$. As before, if one could choose τ for which $|F(x)| > \tau$ only if $\|\Pi_{V\setminus\{w\}}x\|^2 \geq k$, and if we could lower bound $\mathbb{P}[|y| > \tau]$, then we would conclude that $\langle \Pi_{V\setminus\{w\}}, \mathbf{M}_\tau\rangle \geq \mathbb{P}[|y| > \tau]$, so $\mathbf{M}_\tau$ has some other singular vector, orthogonal to w, with non-negligible singular value. The issue is that such a τ typically does not exist! For x satisfying $\|\Pi_{V\setminus\{w\}}x\|^2 \leq k$, $F(x)$ can be arbitrarily large, because $\|\Pi_w x\|$ can be arbitrarily large.

It may be possible to lower bound the quantity in the expression for $\langle \Pi_{V\setminus\{w\}}, \mathbf{M}_\tau\rangle$ using a more refined argument, but for general deep ReLU networks or piecewise linear functions, this seems very challenging. At the very least, one must be careful not to prove something too strong, like showing that $v^\top \mathbf{M}_\tau v$ is non-negligible for *any* unit vector $v \in V$. For instance, even when $L = 0$, it could be that all but one of the rows of $\mathbf{W}_0$ lie in a proper subspace $W \subsetneq V$, and for the remaining row u of $\mathbf{W}_0$, $\|\Pi_{V\setminus W}u\|/\|u\|$ is arbitrarily small. In this case, for v in the direction of $\Pi_{V\setminus W}u$, the quadratic form $v^\top \mathbf{M}_\tau v$ is arbitrarily small, and it would be impossible to recover all of V from a reasonable number of samples.

More generally, any proposed algorithm for learning all of V had better be consistent with the fact that it is impossible to recover the full subspace V within a reasonable number of samples if almost all of the variance of F is explained by some proper subspace $W \subsetneq V$, or equivalently, if the "leftover variance" $\mathbb{E}_x[(F(x) - F(\Pi_W x))^2]$ is negligible. We emphasize that this is a key subtlety that does not manifest in previous works that consider full-rank, well-conditioned weight matrices.

Learning the Full Subspace: Our Approach: We now explain our approach. At a high level, we try to learn orthogonal directions inside the relevant subspace in an iterative fashion. The threshold filter approach above already gives us a single direction in V. Suppose inductively that we've learned some orthogonal vectors $w_1, ..., w_\ell \in V$ spanning a subspace $W \subseteq V$ and want to learn another (note that technically we can only guarantee $w_1, ..., w_\ell$ are approximately within V, but let us temporarily ignore this for the sake of exposition). Motivated by the above consideration regarding "leftover variance," we proceed by a win-win argument: either the leftover variance already satisfies $\mathbb{E}_x[(F(x) - F(\Pi_W x))^2] \leq \varepsilon$ in which case we are already done, or we can learn a new direction via the following crucial modification of the threshold filter.

First, as a thought experiment, consider the following matrix

$$\mathbf{M}_\tau^W \triangleq \Pi_{W^\perp} \mathop{\mathbb{E}}_{x,y}\big[\mathbb{1}[|y - F(\Pi_W x)| > \tau] \cdot (xx^\top - \mathrm{Id})\big]\Pi_{W^\perp}.$$

Note the critical fact that we threshold on $y - F(\Pi_W x)$ as opposed to just on y. As before, it is not hard to show that if this matrix is nonzero, then its singular vectors with nonzero singular value must lie in $\mathbf{W}_0$ and be orthogonal to W; thus giving us a new direction in $\mathbf{W}_0$. We claim that if the leftover variance is non-negligible, then the above matrix will give us a new direction in W.

The intuition behind the above matrix is as follows. Let $V\setminus W$ denote the subspace of V orthogonal to W. We can write $F(x) = F(\Pi_V x) = F(\Pi_W x + \Pi_{V\setminus W}x)$. Now, as F is Lipschitz, we can bound $G(x) = y - F(\Pi_W x) = F(\Pi_W x + \Pi_{V\setminus W}x) - F(\Pi_W x)$ as $|G(x)| \leq \Lambda\|\Pi_{V\setminus W}x\|^2$,

where Λ is the Lipschitz constant of F. In other words, $G(x)$ is bounded over x for which $\|\Pi_{V\setminus W}x\|$ is bounded. Recall that the fact that $F(x)$ is not bounded over such x was the key obstacle to using the original threshold filter approach to learn the full subspace.

The upshot is that for a suitably large τ, the only contribution to the matrix $\mathbf{M}_\tau^W$ should be from inputs x that have large projection in $V \setminus W$. We are now in a position to adapt the analysis lower bounding $\langle \Pi_V, \mathbf{M}_\tau \rangle$ to lower bounding $\langle \Pi_{V\setminus W}, \mathbf{M}_\tau^W \rangle$. In particular, we can apply the aforementioned anti-concentration for piecewise linear functions *to the function* G and argue that, provided the leftover variance $\mathbb{E}_x[(F(x) - F(\Pi_W x))^2] = \mathbb{E}_x[G(x)^2]$ is non-negligible, the top singular vector of $\mathbf{M}_\tau^W$ will give us a new vector in $V \setminus W$.

That being said, an obvious obstacle in implementing the above is that along with not knowing the true subspace $\mathbf{W}_0$, we also don't know the true function F. This precludes us from forming the matrix $\mathbf{M}_\tau^W$ as defined above.

To get around this, we will enumerate over a sufficiently fine net of ReLU networks $\widetilde{F}$ *with relevant subspace* W, one of which will be close to the ReLU network $F(\Pi_W x)$. For each $\widetilde{F}$, we will form the matrix $\widetilde{\mathbf{M}}_\tau^W \triangleq \Pi_{W^\perp} \mathbb{E}_{x,y}\Big[\mathbb{1}[|y - \widetilde{F}(\Pi_W x)| > \tau] \cdot (xx^\top - \mathrm{Id})\Big]\Pi_{W^\perp}$. and output the top singular vector as our new direction only if it has non-negligible singular value.

Arguing soundness, i.e. that this procedure doesn't yield a "false positive" in the form of an erroneous direction lying far from V, is not too hard. However, analyzing completeness, i.e. that this procedure will find *some* new direction, is surprisingly subtle (see Lemma V.7). Formally, we need to argue that if we have an approximation $\widetilde{F}$ to the true F (under some suitable metric), then the corresponding matrix $\widetilde{\mathbf{M}}_\tau^W$ is close to the matrix $\mathbf{M}_\tau^W$. This is further complicated by the fact that ultimately, we will only have access to a subspace W which is *approximately* in V, as every direction we find in our iterative procedure is only guaranteed to *mostly* lie within V.

Our key step in proving this is showing a new stability property of affine thresholds of piecewise linear functions and makes an intriguing connection to *lattice polynomials* in tropical geometry.

Stability of Piecewise Linear Functions: Following the above discussions, to complete our analysis we need to show *stability* of affine thresholds of ReLU networks in the following sense: if $F, \tilde{F} : \mathbb{R}^d \to \mathbb{R}$ are two RELU networks that are close in some structural sense (i.e., under some parametrization), then $\mathbb{E}[\mathbb{1}[|F(x)| > \tau](xx^T - Id)] \approx \mathbb{E}[\mathbb{1}[|\tilde{F}(x)| > \tau](xx^T - Id)]$. A natural way to approach the above is to upper bound $\mathbb{P}[|F(x)| > \tau \wedge |\widetilde{F}(x)| \leq \tau]$. That is, affine thresholds of ReLU networks that are structurally close disagree with low probability.

A natural way to parametrize closeness is to require the weight matrices of the two networks $F, \tilde{F}$ to be close to each other. While such a statement is not too difficult to show for depth two networks (by a union bound over pairs of ReLUs), proving such a statement for general ReLU networks using a direct approach seems quite challenging. We instead look at proving such a statement for a more general class of functions - continuous piecewise-linear functions which allows us to do a certain kind of hybrid argument more naturally.

Concretely, we show that affine thresholds of piecewise-linear functions that are close in some appropriate structural sense disagree with low probability over Gaussian space. We will elaborate upon the notion of structural closeness we consider momentarily, but for now it is helpful to keep in mind that it specializes to L_2 distance for linear functions.

Lemma II.2 (Informal, see Lemma V.3)**.** *Let $F, \widetilde{F} : \mathbb{R}^d \to \mathbb{R}$ be piecewise-linear functions, both consisting of at most m linear pieces, which are "(m, η)-structurally-close" (see Definition IV.9). For any $\tau > 0$,*

$$\mathbb{P}_{x\sim\mathcal{N}(0,\mathrm{Id})}\Big[|F(x)| > \tau \wedge |\widetilde{F}(x)| \leq \tau\Big] \leq O(\eta m^2/\tau). \quad (1)$$

To get a sense for this, suppose $F, \widetilde{F}$ were even close in the sense that the polyhedral regions over which F is linear are *identical* to those over which $\widetilde{F}$ is linear, and furthermore $\mathbb{E}_x[(F(x)-\widetilde{F}(x))^2]^{1/2} \leq \eta$. Then if we take for granted that Lemma II.2 holds when $m = 1$, i.e. when $F, \widetilde{F}$ are linear, it is not hard to show an $O((\eta m/\tau)^c)$ upper bound in (1) under this very strong notion of closeness for some $c < 1$. Because F and $\widetilde{F}$ are L_2-close as functions, for any $t > 0$ we have that with probability $1 - O(\eta^2/t^2)$ the input $x \sim \mathcal{N}(0, \mathrm{Id})$ lies in a polyhedral region for which the corresponding linear functions for F and $\widetilde{F}$ are t-close. By the $m = 1$ case of Lemma II.2, over any one of these at most m regions, the affine thresholds $\mathbb{1}[|F(x)| > \tau]$ and $\mathbb{1}[|F(x)| > \tau]$ disagree with probability $O(t/\tau)$. Union bounding over these regions as well as the event of probability η^2/t^2 that x does not fall in such a polyhedral region, we can upper-bound the left-hand side of (1) by $O(\eta^2/t^2 + mt/\tau)$, and by taking $t = (\eta^2\tau/m)^{1/3}$, we get a bound of $(\eta m^2/\tau)^{2/3}$.

The issues with this are twofold. First, recall the function $\widetilde{F}$ that we want to apply Lemma V.3 to is obtained from some enumeration over a fine net of ReLU networks. As such there is no way to guarantee that the polyhedral regions defining F and $\widetilde{F}$ are exactly the same, making adapting the above argument far more difficult, especially for general ReLU networks.

Second, we stress that the *linear* scaling in $O(\eta)$ in (II.2) is essential. If one suffered any polynomial loss in this bound as in the above argument, then upon applying Lemma II.2 k times over the course of our iterative algorithm for recovering V, we would incur time and sample complexity *doubly exponential* in k. The reason is as follows.

Recall that in the final argument we can only ensure that the directions $w_1, \ldots, w_\ell$ we have found so far are *approximately* within V, and the parameter η will end up scaling with an appropriate notion of subspace distance between W and the true space V. On the other hand, the bound we can show on how far $\widetilde{M}_\tau^W$ deviates from M_τ^W in spectral norm will essentially scale with the right-hand side of (II.2). So if we could only ensure $\widetilde{M}_\tau^W$ and M_τ^W are $O(\eta^c)$-close in spectral norm for $c < 1$, then if we append the top eigenvector of $\widetilde{M}_\tau^W$ to the list of directions $w_1, ..., w_\ell$ we have found so far, the resulting span will only be $O(\eta^c)$-close in subspace distance. Iterating, we would conclude that for the final output of the algorithm to be sufficiently accurate, we would need the error incurred by the very first direction w_1 found to be doubly exponentially small in k!

Lattice Polynomials: It turns out that there is a clean workaround to both issues: passing to the *lattice polynomial* representation for piecewise-linear functions. Specifically, we exploit the following powerful tool:

Theorem II.3 ([58], Theorem 4.1; see Theorem IV.8 below)**.** *If F is continuous piecewise-linear, there exist linear functions $\{g_i\}_{i\in[M]}$ and subsets $\mathcal{I}_1, ..., \mathcal{I}_m \subseteq [M]$ for which*

$$F(x) = \max_{j\in[m]} \min_{i\in\mathcal{I}_j} g_i(x). \tag{2}$$

In fact, our notion of "structural closeness" will be built around this structural result. Roughly speaking, we say two piecewise linear functions are structurally close if they have lattice polynomial representations of the form (2) with the same set of clauses and whose corresponding linear functions are pairwise close in L_2 (see Definition IV.9).

At a high level, Theorem II.3 will then allow us to implement a hybrid argument in the proof of Lemma II.2 and carefully track how the affine threshold computed by a piecewise-linear function changes as we interpolate between F and $\widetilde{F}$. In this way, we end up with the desired linear dependence on η in (II.2).

With Lemma II.2 in hand, we can argue that even with only access to a subspace W approximately within V and with only a function $\widetilde{F}$ that approximates $F(\Pi_W x)$, the top singular vector of $\widetilde{\mathbf{M}}_\tau^W$ mostly lies within V, and we can make progress.

Finally, we remark that as an added bonus, Theorem II.3 also gives us a way to enumerate over general continuous piecewise-linear functions! In this way, we can adapt our algorithm for learning ReLU networks to learning arbitrary piecewise-linear functions, with some additional computational overhead.

Enumerating Over Piecewise-Linear Functions and ReLU Networks: There is in fact one more subtlety to implementing the above approach for ReLU networks and getting singly exponential dependence on k.

First note that whereas one can always enumerate over functions computed by lattice polynomials of the form (2) in time $\exp(\mathrm{poly}(M))$, for ReLU networks of size S this can be as large as doubly exponential in S. Instead, we enumerate over ReLU networks in the naive way, that is, enumerating over the $\exp(O(S))$ many possible architectures and netting over weight matrices with respect to spectral norm, giving us only singly exponential dependence on S.

Here is the subtlety. Obviously two ReLU networks with the same architecture and whose weight matrices are pairwise close in spectral norm will be close in L_2. But how do we ensure that the corresponding lattice polynomials guaranteed by Theorem II.3 are structurally close? In particular, getting anything quantitative would be a nightmare if the clause structure of these lattice polynomials depended in some sophisticated, possibly discontinuous fashion on the precise entries of the weight matrices.

Our workaround is to open up the black box of Theorem II.3 and give a proof for the special case of ReLU networks from scratch. In doing so, we will find out that there are lattice polynomial representations for ReLU networks which only depend on the architecture and the *signs* of the entries of the weight matrices (see full version). In this way, we can guarantee that a moderately fine net will contain a network which is structurally close to the true network.

III. Technical Preliminaries

In this section we collect notation and technical tools that will be useful in the sequel.

A. Miscellaneous Notation and Definitions

We will use $\vee$ and $\wedge$ to denote max and min respectively. We will use $\|\cdot\|_p$ to denote the L_p norm of a vector or of a random variable. When the random variable is given by a function over Gaussian space, e.g. $F(x)$ for $x \sim \mathcal{N}(0, \mathrm{Id})$ and $F : \mathbb{R}^d \to \mathbb{R}$, we use the short-hand $\|F\|_p$ to denote $\mathbb{E}_{x\sim\mathcal{N}(0,\mathrm{Id})}[F(x)^p]^{1/p}$. When $p = 2$, we will omit the subscript. We use $\|\cdot\|_{\mathrm{op}}$ and $\|\cdot\|_F$ to denote operator and Frobenius norms respectively. When we refer to a function as Λ-Lipschitz, unless stated otherwise we mean with respect to L_2.

Given a subspace $V \subset \mathbb{R}^d$, let Π_V denote the orthogonal projector to that subspace. Let $\mathbb{S}_V \subset \mathbb{R}^d$ denote the set of vectors in V of unit norm. When the ambient space $\mathbb{R}^d$ is clear from context, we let $V^\perp$ denote the orthogonal complement of V. For a subspace $W \subseteq V$, we will denote the orthogonal complement of W inside V by $V \backslash W$.

Given $x \in \mathbb{R}$, let $\mathcal{N}(0, 1, x)$ denote the standard Gaussian density's value at x. Let $\mathrm{erfc}(z) \triangleq \mathbb{P}_{g\sim\mathcal{N}(0,1)}[|g| > z]$ (note that we eschew the usual normalization). Let χ^2_m denote the chi-squared distribution with m degrees of freedom.

Recall that we denote the ReLU activation function by $\phi(z) \triangleq \max(z, 0)$.

The following class of functions will be useful for us.

Definition III.1. The set of *lattice polynomials over the reals* is the set of real-valued functions defined inductively as follows: for any $d \geq 1$, any constant real-valued function $\mathbb{R}^d \to \mathbb{R}$ is a lattice polynomial, and any function $h : \mathbb{R}^d \to \mathbb{R}$ which can be written as $h(x) = f(x) \vee g(x)$ or $h(x) = f(x) \wedge g(x)$ for two lattice polynomials $f, g : \mathbb{R}^d \to \mathbb{R}$ is also a lattice polynomial.

Fact III.2 (Elementary anticoncentration)**.** *If Z is a random variable for which $|Z| \leq M$ almost surely, and $\mathbb{E}[Z^2] \geq \sigma^2$, then $\mathbb{P}[|Z| \geq t] \geq \frac{1}{M^2}(\sigma^2 - t^2)$.*

Definition III.3 (Frames)**.** A set of orthonormal vectors $\widetilde{w}_1, ..., \widetilde{w}_\ell$ is a *frame*. Given subspace $V \subset \mathbb{R}^d$, we say that this frame is *ν-nearly within V* if $\|\Pi_V \widetilde{w}_i\| \geq 1 - \nu$ for all i. We will sometimes refer to their span $\widetilde{W}$ as a frame ν-nearly within to V, when the choice of orthonormal basis for $\widetilde{W}$ is clear from context.

IV. Continuous Piecewise-Linear Functions and Lattice Polynomials

In this section, we introduce tools for reasoning about continuous piecewise-linear functions, culminating in a structural result (Theorem IV.8) giving an explicit representation of arbitrary ReLU networks as lattice polynomials (see Definition III.1).

A. Basic Notions

We will work with functions which only depend on some low-dimensional projection of the input.

Definition IV.1 (Subspace juntas)**.** A function $F : \mathbb{R}^d \to \mathbb{R}$ is a *subspace junta* if there exist $v_1, ..., v_k \in \mathbb{S}^{d-1}$ and a function $h : \mathbb{R}^k \to \mathbb{R}$ for which $F(x) = h(\langle v_1, x\rangle, ..., \langle v_k, x\rangle)$ for all $x \in \mathbb{R}^d$. We will refer to $V \triangleq \text{span}(v_1, ..., v_k)$ as the *relevant subspace* of F, to $v_1, ..., v_k$ as the *relevant directions* of F, and to h as the *link function* of F.

Definition IV.2 (Piecewise Linear Functions)**.** Given vector space W, a function $h : W \to \mathbb{R}$ is said to be *piecewise-linear* (resp. piecewise-affine-linear) if there exist finitely many linear (resp. affine linear) functions $\{g_i : W \to \mathbb{R}\}_{i \in [M]}$ and a partition of W into finitely many polyhedral cones $\{S_i\}_{i \in \mathcal{I}}$ such that $G(x) = \sum_i \mathbb{1}[x \in S_i] g_i(x)$. We will say that h is *realized by M pieces* $\{(g_i, S_i)\}$ (note that h can have infinitely many realizations). If each g_i is given by $g_i(x) = \langle u_i, x\rangle + b_i$ for some $u_i \in W, b_i \in \mathbb{R}$, then we will also refer to the pieces of h by $\{(\langle u_i, \cdot\rangle + b_i, S_i)\}$.

We are now ready to define the concept class we will work with in this paper.

Definition IV.3 ("Kickers")**.** We call a subspace junta F with link function h a *kicker* if h is continuous piecewise-linear. Note that a kicker is itself a continuous piecewise-linear function, and for any realization of its link function by M pieces, there is a realization of F by M pieces.

Henceforth, fix a subspace junta $F : \mathbb{R}^d \to \mathbb{R}$ with link function h and relevant directions $v_1, ..., v_k$ spanning relevant subspace $V \subset \mathbb{R}^d$.

Example IV.4 (ReLU Networks)**.** *Feedforward ReLU networks as defined in Definition I.1 are kickers with relevant subspace of dimension at most k, where k is the row span of the weight matrix $\mathbf{W}_0$, the link function is defined by*

$$h(z) = \mathbf{W}_{L+1}\phi(\mathbf{W}_L\phi(\cdots \mathbf{W}_1\phi(z)\cdots)),$$

and the pieces in one possible realization of h correspond to the different possible sign patterns that the activations could take on, that is the different possible values of the vector $\{\mathbf{W}_a\phi(\mathbf{W}_{a-1}\phi(\cdots \mathbf{W}_1\phi(z)\cdots))\}_{0 \leq a \leq L} \in \prod_{a=0}^{L}\{\pm 1\}^{k_a}$ as z ranges over $\mathbb{R}^k$.

Lemma IV.5. *If F is a Λ-Lipschitz kicker, then for any realization of its link function h by pieces $\{(\langle w_i, \cdot\rangle, S_i)\}$, there is a realization by pieces $\{(\langle w_i', \cdot\rangle, S_i)\}$ for which $\max_i \|g_i\| \leq L$..*

Definition IV.6 (Restrictions)**.** Given any nonzero linear subspace $W \subseteq V$, let $F|_W : W \to \mathbb{R}$ denote the restriction of F to the subspace W. By abuse of notation, we will sometimes also regard $F|_W$ as a function over $\mathbb{R}^d$ given by $F|_W(x) = F(\Pi_W x)$.

The following property of restrictions of Lipschitz functions will be important.

Lemma IV.7. *For any nonzero linear subspace $W \subseteq V$, and Λ-Lipschitz function $F : \mathbb{R}^d \to \mathbb{R}$, $\sup_{x:\|\Pi_{V \setminus W} x\| \leq 1} |F(x) - F(\Pi_W x)| \leq \Lambda$.*

B. A Generic Lattice Polynomial Representation

Essential to our analysis is the following structural result from [58] which says that, perhaps surprisingly, *any* piecewise linear function can be expressed as a relatively simple lattice polynomial.

Theorem IV.8 ([58], Theorem 4.1)**.** *If $h : \mathbb{R}^n \to \mathbb{R}$ is a continuous piecewise-linear function which has a realization by pieces $\{(g_i, S_i)\}_{i \in [M]}$, there exists a collection of* clauses *$\mathcal{I}_1, ..., \mathcal{I}_m \subseteq [M]$ for which*

$$h(x) = \max_{j \in [m]} \min_{i \in \mathcal{I}_j} g_i(x) \tag{3}$$

Technically we will need to use this theorem in a white-box fashion as the specific construction exhibited in this theorem will be important in the proof of our main result for learning ReLU networks. We defer these details to the full version.

We will work with the following notion of approximation for such lattice polynomials:

Definition IV.9. Two continuous piecewise-linear functions $G, \widetilde{G} : \mathbb{R}^d \to \mathbb{R}$ are (M, η)-*structurally-close* if there exist linear functions $g_1, ..., g_M$ and $\widetilde{g}_1, ..., \widetilde{g}_M$ and subsets $\mathcal{I}_1, ..., \mathcal{I}_m \subseteq [M]$ for which $G(x) = \max_{j\in[m]} \min_{i\in\mathcal{I}_i} g_i(x)$, $\widetilde{G}(x) = \max_{j\in[m]} \min_{i\in\mathcal{I}_i} \widetilde{g}_i(x)$, and $\|g_i - \widetilde{g}_i\| \le \eta$ for all i.

Structural closeness of continuous piecewise-linear functions in the above sense is stronger than L_2-closeness.

Lemma IV.10. *Take continuous piecewise-linear functions $G, \widetilde{G} : \mathbb{R}^m \to \mathbb{R}$ which are (M, η)-structurally-close. Then $\|G - \widetilde{G}\| \le \eta\sqrt{m}$. In particular, if G is a piecewise-linear function which is realized by pieces $\{(\langle u_i, \cdot\rangle, S_i)\}$ satisfying $\|u_i\| \le \eta$, then $\|G\| \le \eta\sqrt{m}$.*

As discussed in Section II, for our application to learning general kickers, we will leverage the lattice polynomial representation in Theorem IV.8 to grid over piecewise-linear functions. Note that *a priori*, even if we knew exactly the set of linear functions $\{g_i\}_{i\in[M]}$ in a realization of a piecewise-linear function, enumerating over all lattice polynomials of the form (3) would require time doubly exponential in M, as there are 2^M possible clauses $\mathcal{I}_j$ and 2^{2^M} possible sets of clauses $\{\mathcal{I}_j\}$.

By being slightly more careful, we can enumerate in time $\exp(\mathrm{poly}(M))$, see full version.

V. Filtered PCA

In this section we prove our main results on learning kickers and ReLU networks. Throughout, we will make the following base assumption about the function F.

Assumption 1. *F is a kicker which is Λ-Lipschitz for some $\Lambda \ge 1$ and has at most M pieces.*

While our techniques are general enough to work under just this assumption, for our main application to learning ReLU networks (Definition I.1), we can obtain improved runtime guarantees by making the following additional assumption on F.

Assumption 2. *F is computed by a size-S ReLU network[7] with depth $L + 2$ and weight matrices $\mathbf{W}_0 \in \mathbb{R}^{k_0\times d}, \dots \mathbf{W}_L \in \mathbb{R}^{k_L\times k_{L-1}}, \mathbf{W}_{L+1} \in \mathbb{R}^{1\times k_L}$ satisfying $\|\mathbf{W}_i\|_{op} \le B$ for all $0 \le i \le L + 1$, for some $B \ge 1$.*[8]

In this section, unless stated otherwise, we will only assume F satisfies Assumption 1, but in certain parts of the proof, we will get better bounds by additionally making Assumption 2. Under these assumptions, we will prove Theorems I.2 and I.5.

In Section V-A, we prove an anti-concentration result for piecewise-linear functions. We use this to prove that in an idealized scenario where we had exact access to some ℓ-dimensional $W \subset V$ as well as exact query access to $F|_W$, we would be able to approximately recover a vector in $V\backslash W$ by running one iteration of the main loop of FilteredPCA. In the remaining sections, we show how to pass from this idealized scenario to the setting we actually care about, in which we only samples $(x, F(x))$. In Section V-B we show that affine thresholds of piecewise-linear functions are stable under small perturbations of the function. Then in Section V-C, we show how to grid over the set of kickers/ReLU networks and formally state our algorithm. In Section V-D we combine these ingredients to argue that as long as we have sufficiently good approximate access to W and $F|_W$, a single iteration of the main loop of FilteredPCA will approximately recover a vector from $V\backslash W$, from which our main theorems will follow.

A. Anti-Concentration of Piecewise Linear Functions

An important technical tool is the following result showing that for any continuous piecewise-linear function with some variance, the probability that it exceeds any given threshold is non-negligible.

Lemma V.1. *If $G : \mathbb{R}^m \to \mathbb{R}$ is continuous piecewise-linear and Λ-Lipschitz and $\mathbb{E}[G^2] \ge \sigma^2$, then for any $s \ge 0$, $\mathbb{P}[|G| > s] \ge \Omega(\exp(-3ms^2/\sigma^2)) \cdot \frac{s\sigma}{\sqrt{m}\Lambda^2}$.*

Now suppose we had access to an orthonormal collection of vectors $w_1, \dots, w_\ell$ that are *exactly* in V. Let W denote their span. Suppose further that we had access to the matrix

$$\mathbf{M}_\tau^W \triangleq \Pi_{W^\perp} \mathop{\mathbb{E}}_{x,y} \big[\mathbb{1}[|y - F(\Pi_W x)| > \tau] \cdot (xx^\top - \mathrm{Id})\big] \Pi_{W^\perp}.$$

When the threshold τ is clear from context, we will just refer to this matrix as $\mathbf{M}^W$.

As we will see, if this matrix is nonzero, then its singular vectors with nonzero singular value must lie in V and be orthogonal to $w_1, \dots, w_\ell$. The main challenge will be to show that this matrix is nonzero. The following proof also applies to the case of $\ell = 0$, in which case $F(\Pi_W x)$ specializes to the zero function and (4) specializes to

$$\mathbf{M}_\tau^\emptyset \triangleq \mathop{\mathbb{E}}_{x,y} \big[\mathbb{1}[|y| > \tau] \cdot (xx^\top - \mathrm{Id})\big]. \tag{4}$$

In particular, (4) is a matrix we actually have access to at the beginning of the algorithm, and one consequence of the warmup argument below is an algorithm for finding a single vector in V.

It is not hard to show that for appropriately chosen τ, either the top singular value of $\mathbf{M}_\tau^W$ is non-negligible, or $\mathbb{E}[(F(x) - F(\Pi_W x)^2]$ is small, that is, F is already sufficiently well-approximated by the function $F|_W$:

Lemma V.2. *Suppose $\mathbb{E}_{x\sim\mathcal{N}(0,\mathrm{Id})}[(F(x) - F(\Pi_W x))^2] \ge \rho^2$ for some $\rho > 0$. For any $\tau > 0$, if a vector is not in the kernel of $\mathbf{M}_\tau^W$, then it must lie in $V\backslash W$. For $\tau \ge \sqrt{2(k-\ell)} \cdot \Lambda$,*

[7] Note that this implies $M \le 2^S$.

[8] Recall from Definition I.1 that we will refer to the rank of $\mathbf{W}_0$ as k to emphasize that F is a kicker with relevant subspace V of dimension k.

$\left\langle \mathbf{M}_\tau^W, \Pi_{V\setminus W} \right\rangle \geq \Omega\left(e^{-3k\tau^2/\rho^2}\right) \cdot \frac{(k-\ell)\tau\rho}{\sqrt{k}\Lambda^2}$. *In particular, for this choice of* τ*, the top singular vector of* $\mathbf{M}_\tau^W$ *lies in* $V\setminus W$ *and has singular value at least* $\lambda_\tau^{(\ell)} \triangleq \Omega\left(e^{-3k\tau^2/\rho^2}\right) \cdot \frac{\tau\rho}{\sqrt{k}\Lambda^2}$.

If ε is the target L_2 error to which we want to learn F, we will only ever work with $\rho \geq \Omega(\varepsilon)$. In the sequel, we will take $\tau = c\sqrt{k}\cdot\Lambda$ for sufficiently large absolute constant $c > 0$. As a result, we have that $\lambda_\tau^{(\ell)} \geq \Omega\left(e^{-O(k^2\Lambda^2/\varepsilon^2)}\right) \cdot (\varepsilon/\Lambda) \triangleq \underline{\lambda}$.

B. Stability of Piecewise Linear Threshold Functions

To get an iterative algorithm for finding all relevant directions of F, we need to show an analogue of Lemma V.2 in the setting when we only have access to directions $\widetilde{w}_1, \ldots, \widetilde{w}_\ell$ which are *close* to the span of V, and when we only have access to an *approximation* of the function $F|_W$.

To this end, another important tool we show in the full version is the following stability result for affine thresholds of piecewise-linear functions:

Lemma V.3. *Let* $f, g, g' : \mathbb{R}^d \to \mathbb{R}$ *be piecewise-linear functions. For any* $\tau > 0$*, if* g, g' *are* (m, η)*-structurally-close and* f *has a realization with at most* m *pieces, then* $\mathbb{P}_x[|g(x) - f(x)| > \tau \wedge |g'(x) - f(x)| \leq \tau] \leq 9\eta m^2/\tau$.

C. Netting Over Piecewise Linear Functions

Suppose we have recovered an ℓ-dimensional subspace $\widetilde{W}$ that approximately lies within V. In this section we show how to produce a finite list of candidate kickers with relevant subspace $\widetilde{W}$, one of which is guaranteed to approximate F restricted to some ℓ-dimensional subspace W. Ignoring the finiteness of this list for now, we first show that as long as $\widetilde{W}$ is sufficiently close to lying within V, there exists *some* kicker close to *some* restriction $F|_W$.

Lemma V.4. *Let* $\widetilde{w}_1, \ldots, \widetilde{w}_\ell$ *be a frame* ν*-nearly within* V*, with span* $\widetilde{W}$*. There exist an* ℓ*-dimensional subspace* $W \subset V$ *and a* Λ*-Lipschitz kicker* $\widetilde{F}^*$ *with relevant subspace* $\widetilde{W}$ *which is* $(M, 2\sqrt{\nu}\cdot\ell\Lambda)$*-structurally-close to* $F|_W$.

We can show that if we enumerate over a fine enough net of kickers, then we can recover an approximation to $\widetilde{F}^*$ from Lemma V.4 in time singly exponential in $\mathrm{poly}(M)$.

Lemma V.5. *Take any* $\varepsilon' > 0$*. Given a frame* $\widetilde{w}_1, \ldots, \widetilde{w}_\ell$ *with span* $\widetilde{W}$*, for any* Λ*-Lipschitz kicker* $\widetilde{F}^*$ *with relevant subspace* $\widetilde{W}$*, there exists a kicker* $\widetilde{F}$ *with relevant subspace* $\widetilde{W}$ *in the output* $\mathcal{L}$ *of* EnumerateKickers$(\widetilde{W}, \varepsilon')$ *which is* $(M, \varepsilon'\Lambda)$*-structurally-close to* $\widetilde{F}$*. Furthermore,* $|\mathcal{L}| \leq M^{M^2}\cdot(1 + 2/\varepsilon')^\ell$.

In particular, if $\widetilde{w}_1, \ldots, \widetilde{w}_\ell$ *is a frame* ν*-nearly within* V*, then for* $\varepsilon' = 2\sqrt{\nu}\cdot\ell$*,* $\mathcal{L}$ *contains a kicker* $\widetilde{F}$ *which is* $(M, C_{\mathsf{piecewise}}\sqrt{\nu})$*-structurally-close to* $F|_W$ *for some* ℓ*-dimensional subspace* $W \subseteq V$*, where* $C_{\mathsf{piecewise}} \triangleq 4k\Lambda$*. Furthermore,* $|\mathcal{L}| \leq M^{M^2}O(1/\sqrt{\nu})^\ell$ *in this case.*

Enumerating over arbitrary kickers with M pieces requires runtime scaling exponentially in $\mathrm{poly}(M)$. For ReLU networks of size S, M could be as large as $\exp(S)$, so naively using EnumerateKickers in our application to learning ReLU networks would incur doubly exponential dependence on k in the runtime. In the full version we describe how to net more efficiently for ReLU networks.

With subroutines for enumerating over ReLU networks and kickers in hand, we can now formally state our algorithm, FilteredPCA (see Algorithm 1 below). The algorithm as stated applies to the case where F is a neural network satisfying Assumptions 1 and 2, but we can easily modify the algorithm to work in the case where F is only a kicker satisfying Assumption 1 by an appropriate change of parameters.

Algorithm 1: FilteredPCA$(\mathcal{D}, \varepsilon, \delta)$

$\mathcal{W} \leftarrow \emptyset$.
for $0 \leq \ell \leq k-1$ **do**
  Draw samples $(x_1, y_1), \ldots, (x_N, y_N) \sim \mathcal{D}$.
  $\widetilde{W} \leftarrow$ span of vectors in $\mathcal{W}$.
  $\mathcal{L} \leftarrow$ net of networks over $\widetilde{W}$.
  for $\widetilde{F} \in \mathcal{L}$ **do**
    Define $\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}$ to be the matrix given by $\sum_{i=1}^N \mathbb{1}\left[|y_i - \widetilde{F}(\Pi_{\widetilde{W}}x)| > \tau\right] \cdot (x_i x_i^\top - \mathrm{Id})$.
    $\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}} \leftarrow \Pi_{\widetilde{W}^\perp}\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}\Pi_{\widetilde{W}^\perp}$.
    Get top singular vector $\widetilde{w}^{\ell+1}$ of $\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}$.
    if $\|\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}\|$ *sufficiently large* **then**
      Append $\widetilde{w}^{\ell+1}$ to $\mathcal{W}$, exit out of this inner loop, and increment ℓ.
  if *no* $\widetilde{w}^{\ell+1}$ *was appended to* $\mathcal{W}$ **then**
    Break.
$\widetilde{W} \leftarrow$ span of vectors in $\mathcal{W}$.
$\mathcal{L} \leftarrow$ net of networks over $\widetilde{W}$.
for $\widetilde{F} \in \mathcal{L}$ **do**
  Form an empirical estimate $\hat{\varepsilon}$ for $\|\widetilde{F} - F\|$.
  if $\hat{\varepsilon} \leq 3\varepsilon$ **then**
    return $\widetilde{F}$.

D. Perturbation Bounds

We now show how to leverage Lemma V.3 to show that even with access to a subspace $\widetilde{W}$ which is only approximately within V as well as the restriction of F to that subspace, we can recover another vector orthogonal to $\widetilde{W}$ which mostly lies within V.

The first step is to show that in this approximate setting, the analogue of $\mathbf{M}^W$ from Lemma V.2 is spectrally close

to $\mathbf{M}^W$. It is in showing this perturbation bound that we invoke the stability result of Section V-B.

Lemma V.6. *Suppose F only satisfies Assumption 1 (resp. both Assumptions 1 and 2). Let $\widetilde{w}_1, \ldots, \widetilde{w}_\ell \in \mathbb{S}^{d-1}$ be a frame ν-nearly within V, with span $\widetilde{W}$. For $* \in \{\mathsf{piecewise}, \mathsf{network}\}$, define $\xi_*(\nu) \triangleq O\left(k\left(\frac{C_*\sqrt{\nu}M^2}{c\sqrt{k}\Lambda}\right)^{1-1/k} \vee \sqrt{\nu k}\right)$. and suppose $N \geq \Omega(\{d \vee \log(1/\delta)\}/\xi_*^2)$. Given subspace $W \subseteq V$ and $\widetilde{F}$ for which $F|_W$ and $\widetilde{F}$ are $(M, C_{\mathsf{piecewise}}\sqrt{\nu})$-structurally-close (resp. $(M, C_{\mathsf{network}}\sqrt{\nu})$-structurally close), then we have that $\|\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}} - \mathbf{M}^W\|_{op} \leq 3\xi(\nu)$ with probability at least $1-\delta$.*

Finally, we use the above perturbation bound to show that in a single iteration of the main outer loop of FILTEREDPCA, if there is some variance unexplained by the subspace $\widetilde{W}$ found so far, then we will find another "good" direction orthogonal to $\widetilde{W}$ which is also approximately within the span of V. Note that this claim has two components: *completeness*, i.e. in the list of candidate functions we have enumerated, there is *some* function for which the top singular vector of $\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}$ is a good direction, and *soundness*, i.e. whatever direction is ultimately chosen in Step 11 of FILTEREDPCA is a good direction.

Lemma V.7. *Suppose F only satisfies Assumption 1 (resp. both Assumptions 1 and 2). Suppose $\nu \leq \varepsilon^2/(4kC_{\mathsf{piecewise}}^2)$ (resp. $\nu \leq \varepsilon^2/(4kC_{\mathsf{network}}^2)$). For $0 \leq \ell < k$, let $\widetilde{w}_1, \ldots, \widetilde{w}_\ell$ be a frame ν-nearly within V, with span $\widetilde{W}$. Define $\xi = \xi_{\mathsf{piecewise}}(\nu)$ (resp. $\xi = \xi_{\mathsf{network}}(\nu)$) according to Lemma V.6, and suppose $N \geq \Omega(\{d \vee \log(1/\delta)\}/\xi^2)$ and $\tau = c\sqrt{k} \cdot \Lambda$.*

Suppose $\xi \leq \underline{\lambda}/6$, and suppose $\mathbb{E}_{x\sim\mathcal{N}(0,\mathrm{Id})}[(F(x) - F(\Pi_{\widetilde{W}}x))^2] \geq \varepsilon^2$. Let $\mathcal{L}$ be the output of ENUMERATEKICKERS*$(\widetilde{W}, 2\sqrt{\nu} \cdot \ell)$ (resp.* ENUMERATENETWORKS*$(\widetilde{W}, 2\sqrt{\nu} \cdot \ell\sqrt{k} \cdot B)$). With probability at least $1 - |\mathcal{L}| \cdot \delta$ over the randomness of the N samples, the following hold:*

1) ***Completeness**: There exists some $\widetilde{F} \in \mathcal{L}$ such that the top singula value of $\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}$ is at least $\underline{\lambda} - 3\xi$.*
2) ***Soundness**: For any $\widetilde{F} \in \mathcal{L}$ for which $\|\widetilde{\mathbf{M}}_{\mathsf{emp}}^{\widetilde{W}}\|_{op} \geq \underline{\lambda} - 3\xi$, the top singular vector w satisfies $\|\Pi_V w\| \geq 1 - c'\xi^2/\underline{\lambda}^2$ for some absolute constant $c' > 0$ and is orthogonal to $\widetilde{W}$.*

By putting the above ingredients together, we can conclude the proof of Theorems I.2 and I.5, see full version.

REFERENCES

[1] I. Diakonikolas and D. 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)*, 2020, pp. 184–195.

[2] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Conference On Learning Theory*. PMLR, 2018, pp. 297–299.

[3] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv*, pp. arXiv–1506, 2015.

[4] Y. Zhang, J. D. Lee, and M. I. Jordan, "L1-regularized neural networks are improperly learnable in polynomial time," in *33rd International Conference on Machine Learning, ICML 2016*, 2016, pp. 1555–1563.

[5] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 4140–4149.

[6] A. Brutzkus and A. Globerson, "Globally optimal gradient descent for a convnet with gaussian inputs," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 605–614.

[7] S. Goel, V. Kanade, A. Klivans, and J. Thaler, "Reliably learning the relu in polynomial time," in *Conference on Learning Theory*. PMLR, 2017, pp. 1004–1042.

[8] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 597–607.

[9] Q. Zhang, R. Panigrahy, and S. Sachdeva, "Electron-proton dynamics in deep learning," *CoRR*, vol. abs/1702.00458, 2017. [Online]. Available: http://arxiv.org/abs/1702.00458

[10] Y. 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*, vol. 70. PMLR, 2017, pp. 3404–3413.

[11] S. Goel, A. R. Klivans, and R. Meka, "Learning one convolutional layer with overlapping patches," in *ICML*, vol. 80. PMLR, 2018, pp. 1778–1786.

[12] S. S. Du, J. D. Lee, and Y. Tian, "When is a convolutional filter easy to learn?" in *6th International Conference on Learning Representations*, 2018.

[13] R. Ge, J. D. Lee, and T. Ma, "Learning one-hidden-layer neural networks with landscape design," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[14] R. Ge, R. Kuditipudi, Z. Li, and X. Wang, "Learning two-layer neural networks with symmetric inputs," in *International Conference on Learning Representations*, 2018.

[15] P. Manurangsi and D. Reichman, "The computational complexity of training relu (s)," *arXiv preprint arXiv:1810.04207*, 2018.

[16] A. Bakshi, R. Jayaram, and D. P. Woodruff, "Learning two layer rectified neural networks in polynomial time," in *Conference on Learning Theory*. PMLR, 2019, pp. 195–268.

[17] S. Goel and A. R. Klivans, "Learning neural networks with two nonlinear layers in polynomial time," in *Conference on Learning Theory*, 2019, pp. 1470–1499.

[18] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *Advances in neural information processing systems*, 2019, pp. 6158–6169.

[19] S. Vempala and J. Wilmes, "Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds," in *COLT*, vol. 99, 2019.

[20] X. Zhang, Y. Yu, L. Wang, and Q. Gu, "Learning one-hidden-layer relu networks via gradient descent," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1524–1534.

[21] I. Diakonikolas, S. Goel, S. Karmalkar, A. R. Klivans, and M. Soltanolkotabi, "Approximation schemes for relu regression," in *Conference on Learning Theory*, 2020.

[22] W. Gao, A. V. Makkuva, S. Oh, and P. Viswanath, "Learning

one-hidden-layer neural networks under general input distributions," *CoRR*, vol. abs/1810.04133, 2018.
[23] Y. Li, T. Ma, and H. R. Zhang, "Learning over-parametrized two-layer neural networks beyond ntk," in *Conference on Learning Theory 2020*, vol. 125. PMLR, 2020, pp. 2613–2682.
[24] I. Diakonikolas, D. M. Kane, V. Kontonis, and N. Zarifis, "Algorithms and sq lower bounds for pac learning one-hidden-layer relu networks," in *Conference on Learning Theory*, 2020, pp. 1514–1539.
[25] A. Daniely, "Sgd learns the conjugate kernel class of the network," *CoRR*, vol. abs/1702.08503, 2017.
[26] S. Goel, A. Gollakota, Z. Jin, S. Karmalkar, and A. Klivans, "Superpolynomial lower bounds for learning one-layer neural networks using gradient descent," *arXiv preprint arXiv:2006.12011*, 2020.
[27] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in neural information processing systems*, 2018, pp. 8571–8580.
[28] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 33, pp. E7665–E7671, 2018.
[29] A. Blum and R. L. Rivest, "Training a 3-node neural network is np-complete," in *Advances in neural information processing systems*, 1989, pp. 494–501.
[30] V. Vu, "On the infeasibility of training neural networks with small mean-squared error," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2892–2900, 2006.
[31] A. R. Klivans and A. A. Sherstov, "Cryptographic hardness for learning intersections of halfspaces," *Journal of Computer and System Sciences*, vol. 75, no. 1, pp. 2–12, 2009.
[32] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Advances in neural information processing systems*, 2014, pp. 855–863.
[33] L. Song, S. Vempala, J. Wilmes, and B. Xie, "On the complexity of learning neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5520–5528.
[34] S. Shalev-Shwartz, O. Shamir, and S. Shammah, "Failures of gradient-based deep learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 3067–3075.
[35] O. Shamir, "Distribution-specific hardness of learning neural networks," *Journal of Machine Learning Research*, vol. 19, no. 32, pp. 1–29, 2018.
[36] S. Goel, S. Karmalkar, and A. Klivans, "Time/accuracy tradeoffs for learning a relu with respect to gaussian marginals," in *Advances in Neural Information Processing Systems*, 2019, pp. 8584–8593.
[37] A. Daniely and G. Vardi, "Hardness of learning neural networks with natural weights," *arXiv preprint arXiv:2006.03177*, 2020.
[38] S. S. Vempala and Y. Xiao, "Structure from local optima: Learning subspace juntas via higher order pca," *arXiv preprint arXiv:1108.3329*, 2011.
[39] A. De, E. Mossel, and J. Neeman, "Is your function low dimensional?" in *Conference on Learning Theory*, 2019, pp. 979–993.
[40] ——, "Robust testing of low-dimensional functions," *arXiv preprint arXiv:2004.11642*, 2020.
[41] R. Dudeja and D. Hsu, "Learning single-index models in gaussian space," in *Conference On Learning Theory*, 2018, pp. 1887–1930.
[42] K.-C. Li, "On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1025–1039, 1992.
[43] D. R. Brillinger, "A generalized linear model with "gaussian" regressor variables," in *Selected Works of David Brillinger*. Springer, 2012, pp. 589–606.
[44] Y. Plan and R. Vershynin, "The generalized lasso with nonlinear observations," *IEEE Transactions on information theory*, vol. 62, no. 3, pp. 1528–1537, 2016.
[45] A. R. Klivans, P. M. Long, and A. K. Tang, "Baum's algorithm learns intersections of halfspaces with respect to log-concave distributions," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2009, pp. 588–600.
[46] A. R. Klivans, R. O'Donnell, and R. A. Servedio, "Learning geometric concepts via gaussian surface area," in *49th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2008, pp. 541–550.
[47] A. Blum and R. Kannan, "Learning an intersection of k halfspaces over a uniform distribution," in *Theoretical Advances in Neural Computation and Learning*. Springer, 1994, pp. 337–356.
[48] S. S. Vempala, "Learning convex concepts from gaussian distributions with pca," in *2010 IEEE 51st Symposium on Foundations of Computer Science*, 2010, pp. 124–130.
[49] ——, "A random-sampling-based algorithm for learning intersections of halfspaces," *Journal of the ACM (JACM)*, vol. 57, no. 6, pp. 1–14, 2010.
[50] D. Babichev, F. Bach *et al.*, "Slice inverse regression with score functions," *Electronic Journal of Statistics*, vol. 12, no. 1, pp. 1507–1543, 2018.
[51] K.-C. Li, "Sliced inverse regression for dimension reduction," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327, 1991.
[52] Y. S. Tan and R. Vershynin, "Polynomial time and sample complexity for non-gaussian component analysis: Spectral methods," in *Conference On Learning Theory*, 2018, pp. 498–534.
[53] N. Goyal and A. Shetty, "Non-gaussian component analysis using entropy methods," in *Proceedings of 51st ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 840–851.
[54] J. Acharya, I. Diakonikolas, J. Li, and L. Schmidt, "Fast algorithms for segmented regression," in *International Conference on Machine Learning*, 2016, pp. 2878–2886.
[55] A. Daniely and G. Vardi, "From local pseudorandom generators to hardness of learning," *arXiv preprint arXiv:2101.08303*, 2021.
[56] B. Applebaum, "Pseudorandom generators with long stretch and low locality from random local one-way functions," in *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, 2012, p. 805–816.
[57] S. Chen and R. Meka, "Learning polynomials of few relevant dimensions," *arXiv preprint arXiv:2004.13748*, 2020.
[58] S. Ovchinnikov, "Max-min representation of piecewise linear functions," *Contributions to Algebra and Geometry*, vol. 43, no. 1, pp. 297–302, 2002.