LEARNING HIERARCHICAL POLYNOMIALS WITH THREE-LAYER NEURAL NETWORKS

Zihao Wang
Peking University
zihaowang@stu.pku.edu.cn

Eshaan Nichani & Jason D. Lee
Princeton University
{eshnich,jasonlee}@princeton.edu

ABSTRACT

We study the problem of learning hierarchical polynomials over the standard Gaussian distribution with three-layer neural networks. We specifically consider target functions of the form $h = g \circ p$ where $p : \mathbb{R}^d \to \mathbb{R}$ is a degree k polynomial and $g : \mathbb{R} \to \mathbb{R}$ is a degree q polynomial. This function class generalizes the single-index model, which corresponds to $k = 1$, and is a natural class of functions possessing an underlying hierarchical structure. Our main result shows that for a large subclass of degree k polynomials p, a three-layer neural network trained via layerwise gradient descent on the square loss learns the target h up to vanishing test error in $\widetilde{\mathcal{O}}(d^k)$ samples and polynomial time. This is a strict improvement over kernel methods, which require $\widetilde{\Theta}(d^{kq})$ samples, as well as existing guarantees for two-layer networks, which require the target function to be low-rank. Our result also generalizes prior works on three-layer neural networks, which were restricted to the case of p being a quadratic. When p is indeed a quadratic, we achieve the information-theoretically optimal sample complexity $\widetilde{\mathcal{O}}(d^2)$, which is an improvement over prior work (Nichani et al., 2023) requiring a sample size of $\widetilde{\Theta}(d^4)$. Our proof proceeds by showing that during the initial stage of training the network performs feature learning to recover the feature p with $\widetilde{\mathcal{O}}(d^k)$ samples. This work demonstrates the ability of three-layer neural networks to learn complex features and as a result, learn a broad class of hierarchical functions.

1 INTRODUCTION

Deep neural networks have demonstrated impressive empirical successes across a wide range of domains. This improved accuracy and the effectiveness of the modern pretraining and finetuning paradigm is often attributed to the ability of neural networks to efficiently learn input features from data. On "real-world" learning problems posited to be hierarchical in nature, conventional wisdom is that neural networks first learn salient input features to more efficiently learn hierarchical functions depending on these features. This feature learning capability is hypothesized to be a key advantage of neural networks over fixed-feature approaches such as kernel methods (Wei et al., 2020; Allen-Zhu & Li, 2020b; Bai & Lee, 2020).

Recent theoretical work has sought to formalize this notion of a hierarchical function and understand the process by which neural networks learn features. These works specifically study which classes of hierarchical functions can be efficiently learned via gradient descent on a neural network, with a sample complexity improvement over kernel methods or shallower networks that cannot utilize the hierarchical structure. The most common such example is the *multi-index model*, in which the target f^* depends solely on the projection of the data onto a low-rank subspace, i.e $f^*(x) = g(Ux)$ for a projection matrix $U \in \mathbb{R}^{r \times d}$ and unknown link function $g : \mathbb{R}^r \to \mathbb{R}$. Here, a hierarchical learning process simply needs to extract the hidden subspace U and learn the r-dimensional function g. Prior work (Abbe et al., 2022; 2023; Damian et al., 2022; Bietti et al., 2022) shows that two-layer neural networks trained via gradient descent indeed learn the low-dimensional feature Ux, and thus learn multi-index models with an improved sample complexity over kernel methods.

Beyond the multi-index model, there is growing work on the ability of deeper neural networks to learn more general classes of hierarchical functions. (Safran & Lee, 2022; Ren et al., 2023;

Nichani et al., 2023) show that three-layer networks trained with variants of gradient descent can learn hierarchical targets of the form $h = g \circ p$, where p is a simple nonlinear feature such as the norm $p(x) = \|x\|_2$ or a quadratic $p(x) = x^\top Ax$. However, it remains an open question to understand whether neural networks can more efficiently learn a broader class of hierarchical functions.

1.1 Our Results

In this work, we study the problem of learning *hierarchical polynomials* over the standard d-dimensional Gaussian distribution. Specifically, we consider learning the target function $h : \mathbb{R}^d \to \mathbb{R}$, where h is equipped with the hierarchical structure $h = g \circ p$ for polynomials $g : \mathbb{R} \to \mathbb{R}$ and $p : \mathbb{R}^d \to \mathbb{R}$ of degree q and k respectively. This class of functions is a generalization of the single-index model, which corresponds to $k = 1$.

Our main result, Theorem 1, is that for a large class of degree k polynomials p, a three-layer neural network trained via layer-wise gradient descent can efficiently learn the hierarchical polynomial $h = g \circ p$ in $\widetilde{\mathcal{O}}(d^k)$ samples. Crucially, this sample complexity is a significant improvement over learning h via a kernel method, which requires $\widetilde{\Omega}(d^{qk})$ samples (Ghorbani et al., 2021). Our high level insight is that the sample complexity of learning $g \circ p$ is the same as that of learning the feature p, as p can be extracted from the low degree terms of $g \circ p$. Since neural networks learn in increasing complexity (Abbe et al., 2022; 2023; Xu, 2020), such learning process is easily implemented by GD on a three-layer neural network. We verify this insight both theoretically via our layerwise training procedure (Algorithm 1) and empirically via simulations in Section A.

Our proof proceeds by showing that during the initial stage of training the network implements kernel regression in d-dimensions to learn the feature p *even though it only sees* $g \circ p$, and in the next stage implements 1D kernel regression to fit the link function g. This feature learning during the initial stage relies on showing that the low-frequency component of the target function $g \circ p$ is approximately proportional to the feature p, by the "approximate Stein's Lemma" stated in Lemma 2, which is our main technical contribution. This demonstrates that three-layer networks trained with gradient descent, unlike kernel methods, do allow for adaptivity and thus the ability to learn features.

1.2 Related Works

Kernel Methods. Initial learning guarantees for neural networks relied on the Neural Tangent Kernel (NTK) approach, which couples GD dynamics to those of the network's linearization about the initialization (Jacot et al., 2018; Soltanolkotabi et al., 2018; Du et al., 2018; Chizat et al., 2019). However, the NTK theory fails to capture the success of neural networks in practice (Arora et al., 2019; Lee et al., 2020; E et al., 2020). Furthermore, Ghorbani et al. (2021) presents a lower bound showing that for data uniform on the sphere, the NTK requires $\widetilde{\Omega}(d^k)$ samples to learn any degree k polynomial in d dimensions. Crucially, networks in the kernel regime cannot learn features (Yang & Hu, 2021), and hence cannot adapt to low-dimensional structure. An important question is thus to understand how neural networks are able to adapt to underlying structures in the target function and learn salient features, which allow for improved generalization over kernel methods.

Two-layer Neural Networks. Recent work has studied the ability of two-layer neural networks to learn features and as a consequence learn hierarchical functions with a sample complexity improvement over kernel methods. For isotropic data, two-layer neural networks are capable of efficiently learning multi-index models, i.e. functions of the form $f^*(x) = g(Ux)$. Specifically, for Gaussian covariates, Damian et al. (2022); Abbe et al. (2023); Dandi et al. (2023) show that two-layer neural networks learn low-rank polynomials with a sample complexity whose dimension dependence does not scale with the degree of the polynomial, and Bietti et al. (2022); Ba et al. (2022) show two-layer networks efficiently learn single-index models. For data uniform on the hypercube, Abbe et al. (2022) shows learnability of a special class of sparse boolean functions in $\mathcal{O}(d)$ steps of SGD. These prior works rely on layerwise training procedures which learn the relevant subspace in the first stage, and fit the link function g in the second stage. Relatedly, fully connected networks trained via gradient descent on standard image classification tasks have been shown to learn such relevant low-rank features (Lee et al., 2007; Radhakrishnan et al., 2022).

Three-layer Neural Networks. Prior work has also shown that three-layer neural networks can learn certain classes of hierarchical functions. Chen et al. (2020) shows that three-layer networks can

more efficiently learn low-rank polynomials by decomposing the function z^p as $(z^{p/2})^2$. Allen-Zhu et al. (2019) uses a modified version of GD to improperly learn a class of three-layer networks via a second-order variant of the NTK. Safran & Lee (2022) shows that certain ball indicator functions of the form $\mathbf{1}_{\|x\|\geqslant\lambda}$ are efficiently learnable via GD on a three-layer network. They accompany this with a lower bound showing that such targets are not even approximatable by polynomially-sized two-layer networks. Ren et al. (2023) shows that a multi-layer mean-field network can learn the target $\mathrm{ReLU}(1-\|x\|)$. Our work considers a broader class of hierarchical functions and features.

Our work is most similar to Allen-Zhu & Li (2019; 2020a); Nichani et al. (2023). Allen-Zhu & Li (2019) considers learning target functions of the form $p+\alpha g\circ p$ with a three-layer residual network similar our architecture (1). They consider a similar hierarchical learning procedure where the first layer learns p while the second learns g. However Allen-Zhu & Li (2019) can only learn the target up to $O(\alpha^4)$ error, while our analysis shows learnability of targets of the form $g\circ p$, corresponding to $\alpha=\Theta(1)$, up to $o_d(1)$ error. Allen-Zhu & Li (2020a) shows that a deeper network with quadratic activations learns a similar class of hierarchical functions up to arbitrarily small error, but crucially requires α to be $o_d(1)$. We remark that our results do require Gaussianity of the input distribution, while Allen-Zhu & Li (2019; 2020a) hold for a more general class of data distributions. Nichani et al. (2023) shows that a three-layer network trained with layerwise GD, where the first stage consists of a single gradient step, efficiently learns the hierarchical function $g\circ p$ when p is a quadratic, with width and sample complexity $\widetilde{\Theta}(d^4)$. Our Theorem 1 extends this result to the case where p is a degree k polynomial. Furthermore, when p is quadratic, Corollary 1 shows that our algorithm only requires a width and sample complexity of $\widetilde{\Theta}(d^2)$, which matches the information-theoretic lower bound. Our sample complexity improvement for quadratic features relies on showing that running gradient descent for multiple steps can more efficiently extract the feature p during the feature learning stage. Furthermore, the extension to degree k polynomial features relies on a generalization of the approximate Stein's lemma, a key technical innovation of our work.

1.3 Notations

We let $\sum_{i_j}$ denote the sum over increasing sequences $(i_1,\dots i_z)$, i.e $\sum_{i_1<i_2<\cdots<i_z}$. We use $X\lesssim Y$ to denote $X\leqslant CY$ for some absolute positive constant C and $X\gtrsim Y$ is defined analogously. We use $\mathrm{poly}(z_1,\dots,z_p)$ to denote a quantity that depends on $z_1,\dots,z_p$ polynomially. We also use the standard big-O notations: $\Theta(\cdot)$, $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$ to only hide absolute positive constants. In addition, we use $\widetilde{\mathcal{O}}$ and $\widetilde{\Omega}$ to hide higher-order terms, e.g., $\mathcal{O}((\log d)(\log\log d)^2)=\widetilde{\mathcal{O}}(\log d)$ and $\mathcal{O}(d\log d)=\widetilde{\mathcal{O}}(d)$. Let $a\wedge b=\min(a,b)$, $[k]=\{1,2,\dots,k\}$ for $k\in\mathbb{N}$. For a vector v, denote by $\|v\|_p:=(\sum_i|v_i|^p)^{1/p}$ the ℓ^p norm. When $p=2$, we omit the subscript for simplicity. For a matrix A, let $\|A\|$ and $\|A\|_F$ be the spectral norm and Frobenius norm, respectively. We use $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ to denote the maximal and the minimal eigenvalue of a real symmetric matrix. For a vector $w\in\mathbb{R}^R$ and $k\leqslant R$, we use $w_{\leqslant k}\in\mathbb{R}^k$ to denote the first k coordinates of w and $w_{>k}$ to denote the last $R-k$ coordinates of w. That is to say, we can write $w=(w_{\leqslant k},w_{>k})$.

2 Preliminaries

2.1 Problem Setup

Our aim is to learn the target function $h:\mathbb{R}^d\to\mathbb{R}$, where $\mathbb{R}^d$ is the input domain equipped with the standard normal distribution $\gamma:=\mathcal{N}(0,I_d)$. We assume our target has a compositional structure, that is to say, $h=g\circ p$ for some $g:\mathbb{R}\to\mathbb{R}$ and $p:\mathbb{R}^d\to\mathbb{R}$.

Assumption 1. *p is a degree k polynomial with $k\geqslant 2$, and g is a degree q polynomial.*

The degree of h is at most $r:=kq$. We treat k,q as absolute constants, and hide constants that depend only on k,q using big-O notation. We require the following mild regularity condition on the coefficients of g.

Assumption 2. *Denote $g(z)=\sum_{0\leqslant i\leqslant q}g_iz^i$. We assume $\sup_i|g_i|=\mathcal{O}(1)$.*

Three Layer Network. Our learner is a three-layer neural network with a bottleneck layer and residual link. Let m_1,m_2 be the two hidden layer widths, and $\sigma_1(\cdot),\sigma_2(\cdot)$ be two activation func-

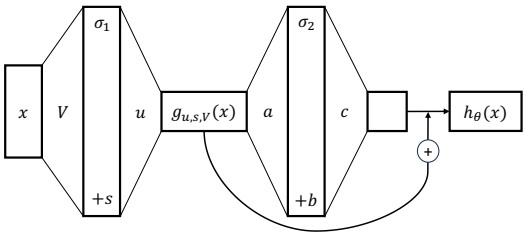

Figure 1: Three-layer network with bottleneck layer and residual link, defined in (1).

tions. The network, denoted by h_θ, is formally defined as follows:

$$h_\theta(x) := g_{u,s,V}(x) + c^\top \sigma_2(a g_{u,s,V}(x) + b) = g_{u,s,V}(x) + \sum_{i=1}^{m_2} c_i \sigma_2(a_i g_{u,s,V}(x) + b_i) \tag{1}$$
$$g_{u,s,V}(x) := u^\top \sigma_1(Vx + s)$$

where $a, b, c \in \mathbb{R}^{m_2}$, $u, s \in \mathbb{R}^{m_1}$ and $V \in \mathbb{R}^{m_1 \times d}$. Here, the intermediate embedding $g_{u,s,V}$ is a two-layer neural network with input x and width m_1, while the mapping $g_{u,s,V} \mapsto h_\theta$ is another two-layer neural network with input dimension 1, width m_2, and a residual connection. We let $\theta := (a, b, c, u, s, V)$ be an aggregation of all the parameters. We remark that the bottleneck layer and residual connection are similar to those in the ResNet architecture (He et al., 2016), as well as architectures considered in prior theoretical work (Ren et al., 2023; Allen-Zhu & Li, 2019; 2020a). See Figure 1 for a diagram of the network architecture.

The parameters $\theta^{(0)} := (a^{(0)}, b^{(0)}, c^{(0)}, u^{(0)}, s^{(0)}, V^{(0)})$ are initialized as $c^{(0)} = 0$, $u^{(0)} = 0$, $a_i^{(0)} \sim_{iid} \mathrm{Unif}\{-1, 1\}$, $s_i^{(0)} \sim_{iid} \mathcal{N}(0, 1/2)$, and $v_i^{(0)} \sim_{iid} \mathrm{Unif}\{\mathbb{S}^{d-1}(1/\sqrt{2})\}$, the sphere of radius $1/\sqrt{2}$, where $\{v_i^{(0)}\}_{i \in [m_1]}$ are the rows of $V^{(0)}$. Furthermore, we will assume $b_i^{(0)} \sim_{iid} \tau_b$, where τ_b is a distribution with density $\mu_b(\cdot)$. We make the following assumption on μ_b:

Assumption 3. *$\mu_b(t) \gtrsim (|t| + 1)^{-p}$ for an absolute constant $p > 0$, and $\mathbb{E}_{b \sim \mu_b}[b^8] \lesssim 1$.*

Remark 1. For example, we can choose τ_b to be the Student's t-distribution with a degree of freedom larger than 8. Student's t-distribution has the probability density function (PDF) given by

$$\mu_\nu(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-(\nu+1)/2}$$

where ν is the number of degrees of freedom and Γ is the gamma function.

Training Algorithm. The network (1) is trained via layer-wise gradient descent with sample splitting. We generate two independent datasets $\mathcal{D}_1, \mathcal{D}_2$, each of which has n independent samples $(x, h(x))$ with $x \sim \gamma$. We denote $\hat{L}_{\mathcal{D}_i}(\theta)$ as the empirical square loss on $\mathcal{D}_i$, i.e

$$\hat{L}_{\mathcal{D}_i}(\theta) := \frac{1}{n} \sum_{x \in \mathcal{D}_i} \left(h_\theta(x) - h(x)\right)^2.$$

In our training algorithm, we first train u via gradient descent for T_1 steps on the empirical loss $\hat{L}_{\mathcal{D}_1}(\theta)$, then train c via gradient descent for T_2 steps on $\hat{L}_{\mathcal{D}_2}(\theta)$. In the whole training process, a, b, s, V are held fixed. The pseudocode for this training procedure is presented in Algorithm 1.

2.2 Hermite Polynomials

Our main results depend on the definition of the Hermite polynomials. We briefly introduce key properties of the Hermite polynomials here, and defer further details to Appendix E.1.

Definition 1 (1D Hermite polynomials)**.** The k-th normalized probabilist's Hermite polynomial, $h_k : \mathbb{R} \to \mathbb{R}$, is the degree k polynomial defined as

$$h_k(x) = \frac{(-1)^k}{\sqrt{k!}} \frac{\frac{d^k \mu_\beta}{dx^k}(x)}{\mu_\beta(x)}, \tag{2}$$

where $\mu_\beta(x) = \exp(-x^2/2)/\sqrt{2\pi}$ is the density of the standard Gaussian.

Algorithm 1 Layer-wise Training Algorithm

Input: Initialization $\theta^{(0)}$, learning rate η_1, η_2, weight decay ξ_1, ξ_2, time T_1, T_2.

for $t = 1, \ldots, T_1$ **do**
  $u^{(t)} \leftarrow u^{(t-1)} - \eta_1(\nabla_u \hat{L}_{\mathcal{D}_1}(\theta^{(t-1)}) + \xi_1 u^{(t-1)})$
  $\theta^{(t)} \leftarrow (a^{(0)}, b^{(0)}, c^{(0)}, u^{(t)}, s^{(0)}, V^{(0)})$
end for
for $t = T_1 + 1, \ldots, T_1 + T_2$ **do**
  $c^{(t)} \leftarrow c^{(t-1)} - \eta_2(\nabla_c \hat{L}_{\mathcal{D}_2}(\theta^{(t-1)}) + \xi_2 c^{(t-1)})$
  $\theta^{(t)} \leftarrow (a^{(0)}, b^{(0)}, c^{(t)}, u^{(T_1)}, s^{(0)}, V^{(0)})$
end for
$\hat{\theta} \leftarrow \theta^{(T_1+T_2)}$

Output: $\hat{\theta}$.

The first such Hermite polynomials are

$$h_0(z) = 1, h_1(z) = z, h_2(z) = \frac{z^2 - 1}{\sqrt{2}}, h_3(z) = \frac{z^3 - 3z}{\sqrt{6}}, \cdots$$

Denote $\beta = \mathcal{N}(0, 1)$ to be the standard Gaussian in 1D. A key fact is that the normalized Hermite polynomials form an orthonormal basis of $L^2(\beta)$; that is $\mathbb{E}_{x\sim\beta}[h_j(x)h_k(x)] = \delta_{jk}$.

The multidimensional analogs of the Hermite polynomials are *Hermite tensors*:

Definition 2 (Hermite tensors)**.** The k-th Hermite tensor in dimension d, $He_k : \mathbb{R}^d \to (\mathbb{R}^d)^{\otimes k}$, is defined as

$$He_k(x) := \frac{(-1)^k}{\sqrt{k!}} \frac{\nabla^k \mu_\gamma(x)}{\mu_\gamma(x)},$$

where $\mu_\gamma(x) = \exp(-\frac{1}{2}\|x\|^2)/(2\pi)^{d/2}$ is the density of the d-dimensional standard Gaussian.

The Hermite tensors form an orthonormal basis of $L^2(\gamma)$; that is, for any $f \in L^2(\gamma)$, one can write the Hermite expansion

$$f(x) = \sum_{k\geqslant 0} \langle C_k(f), He_k(x)\rangle \quad \text{where} \quad C_k(f) := \mathbb{E}_{x\sim\gamma}[f(x)He_k(x)] \in (\mathbb{R}^d)^{\otimes k}.$$

As such, for any integer $k \geqslant 0$ we can define the projection operator $\mathcal{P}_k : L^2(\gamma) \to L^2(\gamma)$ onto the span of degree k Hermite polynomials as follows:

$$(\mathcal{P}_k f)(x) := \langle C_k(f), He_k(x)\rangle.$$

Furthermore, denote $\mathcal{P}_{\leqslant k} := \sum_{0\leqslant i\leqslant k} \mathcal{P}_i$ and $\mathcal{P}_{<k} := \sum_{0\leqslant i<k} \mathcal{P}_i$ as the projection operators onto the span of Hermite polynomials with degree no more than k, and degree less than k, respectively.

3 Main Results

Our goal is to show that the network defined in (1) trained via Algorithm 1 can efficiently learn hierarchical polynomials of the form $h = g \circ p$.

First, we consider a restricted class of degree k polynomials for the hidden feature p. Consider p with the following decomposition:

$$p(x) = \frac{1}{\sqrt{L}} \left(\sum_{i=1}^{L} \lambda_i \psi_i(x) \right). \tag{3}$$

Assumption 4. *The feature p can be written in the form (3). We make the following additional assumptions on p:*

- *There is a set of orthogonal vectors* $\{v_{i,j}\}_{i\in[L],j\in[J_i]}$*, satisfying* $J_i \leqslant k$ *and* $\|v_{i,j}\| = 1$*, such that* $\psi_i(x)$ *only depends on* $v_{i,1}^\top x, \ldots, v_{i,J_i}^\top x$*.*
- *For each* i*,* $\mathcal{P}_k\psi_i = \psi_i$*. Equivalently,* ψ_i *lies in the span of degree* k *Hermite polynomials.*
- $\mathbb{E}\left[\psi_i(x)^2\right] = 1$ *and* $\mathbb{E}\left[p(x)^2\right] = 1$*.*
- *The* λ_i *are balanced, i.e* $\sup_i |\lambda_i| = \mathcal{O}(1)$*, and* $L = \Theta(d)$*.*

Remark 2. The first assumption tells us that each ψ_i depends on a different rank $\leqslant k$ subspace, all of which are orthogonal to each other. As a consequence of the rotation invariance of the Gaussian, the quantities $\psi_i(x)$ are thus independent when we regard x as a random vector. The second assumption requires p to be a degree k polynomial orthogonal to lower-degree polynomials, while the third is a normalization condition. The final condition requires p to be sufficiently spread out, and depend on many ψ_i. Our results can easily be extended to any $L = \omega_d(1)$, at the expense of a worse error floor.

Remark 3. Since $\mathcal{P}_k\psi_i = \psi_i$ for each i, we have $\mathcal{P}_k p = p$. We can thus write $p(x)$ as $\langle A, He_k(x)\rangle$ for some $A \in (\mathbb{R}^d)^{\otimes k}$. There are two important classes of A which satisfy Assumption 4:

First, let A be an orthogonally decomposable tensor

$$A = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i v_i^{\otimes k}\right)$$

where $\langle v_i, v_j\rangle = \delta_{ij}$. Using identities for the Hermite polynomials (Appendix E.1), one can rewrite the feature p as

$$p(x) = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i \langle v_i^{\otimes k}, He_k(x)\rangle\right) = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i h_k(v_i^\top x)\right). \tag{4}$$

p thus satisfies Assumption 4 with $J_i = 1$ for all i, assuming the regularity conditions hold.

Next, we show that Assumption 4 is met when p is a sum of sparse parities, i.e.,

$$A = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i \cdot v_{i,1} \otimes \cdots \otimes v_{i,k}\right)$$

where $\langle v_{i_1,j_1}, v_{i_2,j_2}\rangle = \delta_{i_1 i_2}\delta_{j_1 j_2}$. In that case, the feature p can be rewritten as

$$p(x) = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i \langle v_{i,1} \otimes \cdots \otimes v_{i,k}, He_k(x)\rangle\right) = \frac{1}{\sqrt{L}}\left(\sum_{i=1}^{L} \lambda_i \left(\prod_{j=1}^{k} \langle v_{i,j}, x\rangle\right)\right)$$

For example, taking $L = d/k$ and choosing $v_{i,j} = e_{k(i-1)+j}$, the standard basis elements in $\mathbb{R}^d$, the feature p becomes

$$p(x) = \frac{1}{\sqrt{d/k}}\left(\lambda_1 x_1 x_2 \cdots x_k + x_{k+1}\cdots x_{2k} + \lambda_{d/k} x_{d-k+1}\cdots x_d\right)$$

and hence the name "sum of sparse parities." This feature satisfies Assumption 4 with $J_i = k$ for all i, assuming that the regularity conditions hold.

We next require the following mild assumptions on the link function g and target h. The assumption on h is purely for technical convenience and can be achieved by a simple pre-processing step. The assumption on g, in the single-index model literature (Arous et al., 2021), is referred to as g having an *information exponent* of 1.

Assumption 5. $\mathbb{E}_{x\sim\gamma}\left[h(x)\right] = 0$ *and* $\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right] = \Theta(1)$*.*

Finally, we make the following assumption on the activation functions σ_1, σ_2:

Assumption 6. *We assume* σ_1 *is a* k *degree polynomial. Denote* $\sigma_1(z) = \sum_{0\leqslant i\leqslant k} o_i z^i$*, we further assume* $\sup_i |o_i| = \mathcal{O}(1)$ *and* $|o_k| = \Theta(1)$*. Also, set* $\sigma_2(z) = \max\{z, 0\}$*, i.e., the* ReLU *activation.*

With our assumptions in place, we are ready to state our main theorem.

Theorem 1. *Under the above assumptions, for any constant $\alpha \in (0,1)$, any $m_1 \geqslant d^{k+\alpha}$ and any $n \geqslant d^{k+3\alpha}$, set $m_2 = d^\alpha$, $T_1 = \mathrm{poly}(d, m_1, n)$, $T_2 = \mathrm{poly}(d, m_1, m_2, n)$, $\eta_1 = \frac{1}{\mathrm{poly}(d,m_1,n)}$, $\eta_2 = \frac{1}{\mathrm{poly}(d,m_1,m_2,n)}$, $\xi_1 = \frac{2m_1}{d^{k+\alpha}}$ and $\xi_2 = 2$. Then, for any absolute constant $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of initialization and the sampling of training dataset $\mathcal{D}_1, \mathcal{D}_2$, the estimator $\hat{\theta}$ output by Algorithm 1 satisfies*

$$\left\|h_{\hat{\theta}} - h\right\|^2_{L^2(\gamma)} = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

Theorem 1 states that Algorithm 1 can learn the target $h = g \circ p$ in $n = \widetilde{\mathcal{O}}(d^k)$ samples, with widths $m_1 = \widetilde{\Theta}(d^k), m_2 = \widetilde{\Theta}(1)$. Up to log factors, this is the same sample complexity as directly learning the feature p. On the other hand, kernel methods such as the NTK require $n = \widetilde{\Omega}(d^{kq})$ samples to learn h, and are unable to take advantage of the underlying hierarchical structure.

A simple corollary of Theorem 1 follows when $k = 2$. In this case the feature p is a quadratic polynomial and can be expressed as the following for some symmetric $A \in \mathbb{R}^{d\times d}$

$$p(x) = \langle A, xx^\top - I\rangle = x^\top Ax - \mathrm{tr}(A).$$

Taking $\mathrm{tr}(A) = 0$, and noting that since A always has an eigendecomposition, Assumption 4 is equivalent to $\|A\|_F = 1$ and $\|A\|_{op} = \mathcal{O}(1/\sqrt{d})$, one obtains the following:

Corollary 1. *Let $h(x) = g(x^\top Ax)$ where $\mathrm{tr}(A) = 0, \|A\|_F = 1$, and $\|A\|_{op} = \mathcal{O}(1/\sqrt{d})$. Then under the same setting of hyperparameters as Theorem 1, for any sample size $n \geqslant d^{2+3\alpha}$, with probability at least $1-\delta$ over the initialization and data, the estimator $\hat{\theta}$ satisfies*

$$\left\|h_{\hat{\theta}} - h\right\|^2_{L^2(\gamma)} = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

Corollary 1 states that Algorithm 1 can learn the target $g(x^\top Ax)$ in $\widetilde{\mathcal{O}}(d^2)$ samples, which matches the information-theoretically optimal sample complexity. This improves over the sample complexity of the algorithm in Nichani et al. (2023) when g is a polynomial, which requires $\widetilde{\Theta}(d^4)$ samples. See Section 5.1 for discussion on why Algorithm 1 is able to obtain this sample complexity improvement.

4 Proof Sketch

The proof of Theorem 1 proceeds by analyzing each of the two stages of training. First, we show that after the first stage, the network learns to extract the hidden feature p out (Section 4.1). Next, we show that during the second stage, the network learns the link function g (Section 4.2).

4.1 Stage 1: Feature Learning

The first stage of training is the feature learning stage. Here, the network learns to extract the degree k polynomial feature so that the intermediate layer satisfies $g_{u,s,V} \approx p$ (up to a scaling constant).

At initialization, the network satisfies $h_\theta = g_{u,s,V}$. Thus during the first stage of training, the network trains u to fit $g_{u,s,V}$ to the target h. Since the activation σ_1 is a degree k polynomial with $o_k = \Theta(1)$, we can indeed prove that at the end of the first stage $g_{u,s,V}$ will learn to fit the best degree k polynomial approximation to h, $\mathcal{P}_{\leqslant k}h$ (Lemma 9). During the first stage the loss is convex in u, and thus optimization and generalization can be handled via straightforward kernel arguments. The following lemma formalizes the above argument, and shows that at the end of the first stage the network learns to approximate $\mathcal{P}_{\leqslant k}h$.

Lemma 1. *For any constant $\alpha \in (0,1)$, any $m_1 \geqslant d^{k+\alpha}$ and any $n \geqslant d^{k+3\alpha}$, set $T_1 = \mathrm{poly}(n, m_1, d)$, $\eta_1 = \frac{1}{\mathrm{poly}(n,m_1,d)}$ and $\xi_1 = \frac{2m_1}{d^{k+\alpha}}$. Then, for any absolute constant $\delta \in (0,1)$, with probability at least $1-\delta/2$ over the initialization V, s and training data $\mathcal{D}_1$, we have*

$$\|h_{\theta^{(T_1)}} - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

It thus suffices to analyze the quantity $\mathcal{P}_{\leqslant k}h$. Our key technical result, and a main innovation of our paper, is Lemma 2. It shows that the term $\mathcal{P}_{\leqslant k}h$ is approximately equal to $\mathcal{P}_k h$, and furthermore, up to a scaling constant, $\mathcal{P}_k h$ is approximately equal to the hidden feature p:

Lemma 2. *Under the previous assumptions, we have*

$$\left\|\mathcal{P}_k h - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right]p\right\|_{L^2(\gamma)} = \mathcal{O}(d^{-1/2}) \quad \textit{and} \quad \left\|\mathcal{P}_{<k} h\right\|_{L^2(\gamma)} = \mathcal{O}(d^{-1/2})$$

A proof sketch of Lemma 2 is deferred to Section 4.3, with the full proof in Appendix B.

Combining Lemma 1 and Lemma 2, we obtain the performance after the first stage:

Corollary 2. *Under the setting of hyperparameters in Theorem 1, for any constants $\alpha, \delta \in (0,1)$, with probability $1-\delta/2$ over the initialization and the data $\mathcal{D}_1$, the network after time T_1 satisfies*

$$\left\|h_{\theta^{(T_1)}} - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right]p\right\|^2_{L^2(\gamma)} = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

Proofs for stage 1 are deferred to Appendix C.

4.2 Stage 2: Learning the Link Function

After the first stage of training, $g_{u,s,V}$ is approximately equal to the true feature p up to a scaling constant. The second stage of training uses this feature to learn the link function g. Specifically, the second stage aims to fit the function g using the two-layer network $z \mapsto z + c^\top \sigma_2(az+b)$. Since only c is trained during stage 2, the network is a random feature model and the loss is convex in c.

Our main lemma for stage 2 shows that there exists c^* with low norm such that the parameter vector $\theta^* := (a^{(0)}, b^{(0)}, c^*, u^{(T_1)}, s^{(0)}, V^{(0)})$ satisfies $h_{\theta^*} \approx h$. Let $\hat{p}$ be an arbitrary degree k polynomial satisfying $\left\|\hat{p} - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right]p\right\|^2_{L^2(\gamma)} = \mathcal{O}((\log d)^{r/2} d^{-\alpha})$ (and recall that after stage 1, $g_{u,s,V}$ satisfies this condition with high probability). The main lemma is the following.

Lemma 3. *Let $m = d^\alpha$. With probability at least $1-\delta/4$ over the sampling of a, b, there exists some c^* such that $\|c^*\|_\infty = \mathcal{O}((\log d)^{k(p+q)} d^{-\alpha})$ and*

$$L(\theta^*) = \left\|\hat{p}(x) + \sum_{i=1}^m c_i^* \sigma(a_i \hat{p}(x) + b_i) - h(x)\right\|^2_{L^2(\gamma)} = \mathcal{O}((\log d)^{r/2+2k(p+q)} d^{-\alpha})$$

Since the regularized loss is strongly convex in c, GD converges linearly to some $\hat{\theta}$ with $\hat{L}_2(\hat{\theta}) \lesssim \hat{L}_2(\theta^*)$ and $\|\hat{c}\|_2 \lesssim \|c^*\|_2$. Finally, we invoke standard kernel Rademacher arguments to show that, since the link function g is one-dimensional, $n = \widetilde{\mathcal{O}}(1)$ sample suffice for generalization in this stage. Combining everything yields Theorem 1. Proofs for stage 2 are deferred to Appendix D.

4.3 The Approximate Stein's Lemma

To conclude the full proof of Theorem 1, it suffices to prove Lemma 2. Lemma 2 can be interpreted as an approximate version of Stein's lemma, generalizing the result in Nichani et al. (2023) to polynomials of degree $k > 2$. To understand this intuition, we first recall Stein's lemma:

Lemma 4 (Stein's Lemma)**.** *For any $g : \mathbb{R} \to \mathbb{R}$ and $g \in C^1$, one has*

$$\mathbb{E}_{z\sim\mathcal{N}(0,1)}[zg(z)] = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[g'(z)].$$

Recall that the feature is of the form $p(x) = \frac{1}{\sqrt{L}}\sum_{i=1}^L \lambda_i \psi_i(x)$. Since each ψ_i depends only on the projection of x onto $\{v_{i,1}, \ldots, v_{i,J_i}\}$, and these vectors are orthonormal, the individual terms $\psi_i(x)$ are independent random variables. Furthermore they satisfy $\mathbb{E}[\psi_i(x)] = 0$ and $\mathbb{E}[\psi_i(x)^2] = 1$. Since $L = \Theta(d)$, the Central Limit Theorem tells us that in the $d \to \infty$ limit

$$\frac{1}{\sqrt{L}}\sum_{i=1}^L \lambda_i \psi_i \to_d \mathcal{N}(0,1)$$

when the λ_i are balanced. The distribution of the feature p is thus "close" to a Gaussian. As a consequence, one expects that

$$\mathbb{E}_{x\sim\gamma}[p(x)g(p(x))] \approx \mathbb{E}_{z\sim\mathcal{N}(0,1)}[zg(z)] = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[g'(z)]. \tag{5}$$

Next, let q be another degree k polynomial such that $\|q\|_{L^2(\gamma)} = 1$ and $\langle p, q\rangle_{L^2(\gamma)} = 0$. For most q, we can expect that (p, q) is approximately jointly Gaussian. In this case, p and q are approximately independent due to $\langle p, q\rangle_{L^2(\gamma)} = 0$, and as a consequence

$$\mathbb{E}_{x\sim\gamma}[q(x)g(p(x))] \approx \mathbb{E}_{x\sim\gamma}[q(x)]\,\mathbb{E}_{x\sim\gamma}[g(p(x))] = 0. \tag{6}$$

(5) and (6) imply that the degree k polynomial $g \circ p$ has maximum correlation with is p, and thus

$$\mathcal{P}_k(g \circ p) \approx \mathbb{E}_{z\sim\mathcal{N}(0,1)}[g'(z)]p.$$

Similarly, if q is a degree $< k$ polynomial, then since $\mathcal{P}_k p = p$ one has $\langle p, q\rangle_{L^2(\gamma)} = 0$. Again, we can expect that p, q are approximately independent, which implies that $\langle h, q\rangle_{L^2(\gamma)} \approx 0$.

We remark that the preceding heuristic argument, and in particular the claim that p and q are approximately independent, is simply to provide intuition for Lemma 2. The full proof of Lemma 2, provided in Appendix B, proceeds by expanding the polynomial $g \circ p$ into sums of products of monomials, and carefully analyzes the degree k projection of each of the terms.

5 Discussion

5.1 Comparison to Nichani et al. (2023)

In the case where $k = 2$ and the feature is a quadratic, Corollary 1 tells us that Algorithm 1 requires $\widetilde{\mathcal{O}}(d^2)$ samples to learn h, which matches the information-theoretic lower bound. This is an improvement over Nichani et al. (2023), which requires $\widetilde{\Theta}(d^4)$ samples.

The key to this sample complexity improvement is that our algorithm runs GD for *many steps* during the first stage to completely extract the feature $p(x)$, whereas the first stage in Nichani et al. (2023) takes a single large gradient step, which can only weakly recover the true feature. Specifically, Nichani et al. (2023) considers three-layer neural networks of the form $h_\theta(x) = a^\top\sigma_2(W\sigma_1(Vx) + b)$, and shows that after the first large step of GD on the population loss, the network satisfies $w_i^\top\sigma_1(Vx) \approx d^{-2}p(x)$. As a consequence, due to standard $1/\sqrt{n}$ concentration, $n = \widetilde{\Omega}(d^4)$ samples are needed to concentrate this term and recover the true feature.

On the other hand, the first stage of Algorithm 1 directly fits the best degree 2 polynomial to the target. It thus suffices to uniformly concentrate the loss landscape, which only requires $\widetilde{\mathcal{O}}(d^2)$ samples as the learner is fitting a quadratic. Running GD for many steps is thus key to obtaining this optimal sample complexity. We remark that Nichani et al. (2023) handles a slightly larger class of link functions g (1-Lipschitz functions) and activations σ_1 (nonzero second Hermite coefficient).

5.2 Layerwise Gradient Descent on Three-Layer Networks

Algorithm 1 takes advantage of the underlying hierarchical structure in h to learn in $\widetilde{\Theta}(d^k)$ samples. Regular kernel methods, however, cannot utilize this hierarchical structure, and thus require $\widetilde{\Theta}(d^{kq})$ samples to learn h up to vanishing error. Each stage of Algorithm 1 implements a kernel method: stage 1 uses kernel regression to learn p in $\widetilde{\mathcal{O}}(d^k)$ samples, while stage 2 uses kernel regression to learn g in $\widetilde{\mathcal{O}}(1)$ samples. Crucially, however, *our overall algorithm is not a kernel method*, and can learn hierarchical functions with a significantly improved sample complexity over naively using a single kernel method to learn the entire function. It is a fascinating question to understand which other tasks can be learned more efficiently via such layerwise GD. While Algorithm 1 is layerwise, and thus amenable to analysis, it still reflects the ability of three-layer networks in practice to learn hierarchical targets; see Appendix A for experiments with more standard training procedures.

5.3 Future Work

In this work, we showed that three-layer neural networks are able to efficiently learn hierarchical polynomials of the form $h = g \circ p$, for a large class of degree k polynomials p. An interesting direction is to understand whether our results can be generalized to *all* degree k polynomials. We conjecture that our results should still hold as long as p is homogeneous and close in distribution to a Gaussian, which should be true for more general tensors A. Additionally, the target functions we consider depend on only a single hidden feature p. It is interesting to understand whether deep networks can efficiently learn targets that depend on multiple features, i.e. of the form $h(x) = g(p_1(x), \ldots, p_R(x))$ for some $g : \mathbb{R}^R \to \mathbb{R}$.

ACKNOWLEDGEMENTS

Zihao Wang is partially supported by the elite undergraduate training program of School of Mathematical Sciences at Peking University. Eshaan Nichani acknowledges support from a National Defense Science & Engineering Graduate Fellowship. Eshaan Nichani and Jason D. Lee acknowledge support of the ARO under MURI Award W911NF-11-1-0304, the Sloan Research Fellowship, NSF CCF 2002272, NSF IIS 2107304, NSF CIF 2212262, ONR Young Investigator Award, and NSF CAREER Award 2144994.

REFERENCES

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pp. 4782–4887. PMLR, 2022.

Emmanuel Abbe, Enric Boix Adserà, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 2552–2623. PMLR, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

Zeyuan Allen-Zhu and Yuanzhi Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020a.

Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels?, 2020b.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Gerard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *The Journal of Machine Learning Research*, 22(1):4788–4838, 2021.

Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems*, 35:37932–37946, 2022.

Yu Bai and Jason D. Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks, 2020.

Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. *Advances in Neural Information Processing Systems*, 35:9768–9783, 2022.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Minshuo Chen, Yu Bai, Jason D Lee, Tuo Zhao, Huan Wang, Caiming Xiong, and Richard Socher. Towards understanding hierarchical learning: Benefits of neural representations. *Advances in Neural Information Processing Systems*, 33:22134–22145, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pp. 5413–5452. PMLR, 2022.

Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*, 2023.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Weinan E, Chao Ma, and Lei Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics*, 63(7):1235–1258, jan 2020. doi: 10.1007/s11425-019-1628-5. URL https://doi.org/10.1007%2Fs11425-019-1628-5.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029 – 1054, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area v2. volume Vol 20, 01 2007.

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pp. 3351–3418. PMLR, 2021.

Eshaan Nichani, Alex Damian, and Jason D Lee. Provable guarantees for nonlinear feature learning in three-layer neural networks. *arXiv preprint arXiv:2305.06986*, 2023.

Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

Giuseppe Da Prato and Luciano Tubaro. Wick powers in stochastic pdes: an introduction. 2007. URL https://api.semanticscholar.org/CorpusID:55493217.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Feature learning in neural networks and kernel machines that recursively learn features. *arXiv preprint arXiv:2212.13881*, 2022.

Yunwei Ren, Mo Zhou, and Rong Ge. Depth separation with multilayer mean-field networks. *arXiv preprint arXiv:2304.01063*, 2023.

Itay Safran and Jason Lee. Optimization-based separations for neural networks. In *Conference on Learning Theory*, pp. 3–64. PMLR, 2022.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel, 2020.

Zhi-Qin John Xu. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, jun 2020. doi: 10.4208/cicp.oa-2020-0085. URL https://doi.org/10.4208%2Fcicp.oa-2020-0085.

Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pp. 11727–11737. PMLR, 2021.

Appendix

A Experiments

We empirically verify Theorem 1, and demonstrate that three-layer neural networks indeed learn hierarchical polynomials $g \circ p$ by learning to extract the feature p.

Our experimental setup is as follows. The target feature is of the form $h = g \circ p$, $p(x) = \sum_{i=1}^{d} \lambda_i h_3(x_i)$, where the λ_i are drawn i.i.d from $\left\{\pm\frac{1}{\sqrt{d}}\right\}$ uniformly, and the link function is $g(z) = C_d z^3$, where C_d is a normalizing constant chosen so $\mathbb{E}_x[h(x)^2] = 1$. Our architecture is the same ResNet-like architecture defined in (1), with activations $\sigma_1(z) = z^3$ and $\sigma_2 = \mathrm{ReLU}$. We additionally use the μP initialization (Yang & Hu, 2021). For a chosen input dimension d and sample size n, we choose hidden layer widths $m_1 = d^2$ and $m_2 = 1000$. We optimize the empirical square loss to convergence by simultaneously training all parameters (u, s, V, a, b, c) using the Adam optimizer. We then compute the test loss of the learned predictor, as well as the correlation between the "learned feature" (defined to be $g_{u,s,V}$) and the "true feature" p on these test points.

In Figure 2, we plot both the test loss and feature correlation as a function of n, for $d \in \{16, 24, 32, 40\}$. We observe that, across varying values of depth, roughly d^3 samples are needed to

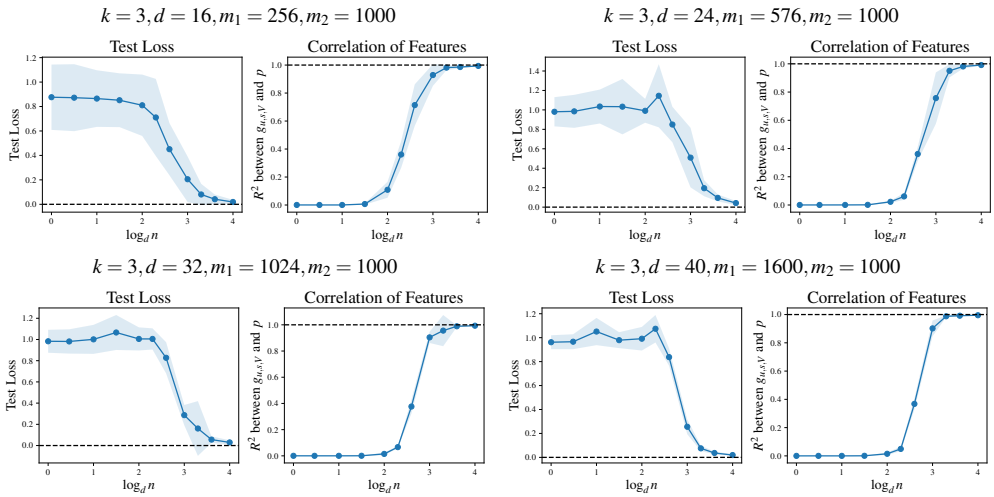

Figure 2: We train the ResNet architecture (1) to learn the hierarchical polynomial $h = g \circ p$ when the degree of p is $k = 3$. We observe that the network learns the true feature p, as measured by the correlation between $g_{u,s,V}$ and p (right panel of each figure). As a consequence, the network can learn h in d^3 samples (left panel of each figure).

learn h up to near zero test error. Additionally, we observe that as n grows past d^3, the correlation between the true feature and learned feature approaches 1. This demonstrates that the network is indeed performing feature learning, and learns to fit p using $g_{u,s,V}$ in order to learn the entire function. Overall, this demonstrates that our high-level insight that the sample complexity of learning $g \circ p$ is equal to the sample complexity of p, and that three-layer neural networks implement the more efficient algorithm of learning to first extract p out of $g \circ p$, holds in the more realistic setting where all parameters of the network are trained jointly.

Experimental Details. Our experiments were written in JAX (Bradbury et al., 2018) and run on a single NVIDIA RTX A6000 GPU.

B Proof of Lemma 2

B.1 Results for General Features

In this subsection, we will consider the following feature class

$$p(x) = \frac{1}{\sqrt{L}} \left(\sum_{i=1}^{L} \lambda_i \psi_i(x) \right)$$

Recall our assumptions on p:

Assumption 4. *The feature p can be written in the form (3). We make the following additional assumptions on p:*

- *There is a set of orthogonal vectors $\{v_{i,j}\}_{i \in [L], j \in [J_i]}$, satisfying $J_i \leqslant k$ and $\|v_{i,j}\| = 1$, such that $\psi_i(x)$ only depends on $v_{i,1}^\top x, \ldots, v_{i,J_i}^\top x$.*
- *For each i, $\mathcal{P}_k \psi_i = \psi_i$. Equivalently, ψ_i lies in the span of degree k Hermite polynomials.*
- $\mathbb{E}\left[\psi_i(x)^2\right] = 1$ *and* $\mathbb{E}\left[p(x)^2\right] = 1$.
- *The λ_i are balanced, i.e* $\sup_i |\lambda_i| = \mathcal{O}(1)$, *and* $L = \Theta(d)$.

Next, recall that the link function $g(z) = \sum_{0 \leqslant i \leqslant q} g_i z^i$ satisfies $\sup_i |g_i| = \mathcal{O}(1)$ by Assumption 2. Denote $h = g \circ p$. Due to Assumption 5, we naturally have $\mathcal{P}_0 h = \mathbb{E}_{x \sim \gamma}[h(x)] = 0$. Next, we will prove the following two Lemmas, which directly implies Lemma 2.

Lemma 5. *Under all the assumptions above, we have*

$$\left\|\mathcal{P}_k h - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right]p\right\|_{L^2(\gamma)} = \mathcal{O}(L^{-1/2})$$

Lemma 6. *Under all the assumptions above, for any $1 \leqslant m \leqslant k-1$ we have*

$$\left\|\mathcal{P}_m h\right\|_{L^2(\gamma)} = \mathcal{O}(L^{-1/2})$$

Proof of Lemma 5. Firstly, we will compute the Hermite degree k components of $p(x)^w$, $w \geqslant 2$. From the definition of $\mathcal{P}_k$ and multinomial expansion theorem, we know

$$\begin{aligned}\mathcal{P}_k(p(x)^w) = \frac{1}{L^{w/2}}\left(\sum_{i=1}^{L}\lambda_i\psi_i(x)\mathcal{P}_0\left(\sum_{z_i\geqslant 2,q,z_1+\cdots+z_q=w-1,i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right)\\ + \frac{1}{L^{w/2}}\mathcal{P}_k\left(\sum_{z_i\geqslant 2,q,z_1+\cdots+z_q=w,i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\end{aligned} \tag{7}$$

by expanding $\left(\frac{1}{\sqrt{L}}\left(\sum_{1\leqslant i\leqslant L}\lambda_i\psi_i(x)\right)\right)^w$ and computing the projection for each term. The key observation that leads to (7) is the following:

Lemma 7. *Let $\phi_1, \phi_2 \in L^2(\gamma)$ be two functions such that ϕ_1 lies in the span of degree k_1 Hermite polynomials and ϕ_2 lies in the span of degree k_2 Hermite polynomials. That is to say, $\mathcal{P}_{k_i}\phi_i = \phi_i$ for $i = 1, 2$.*

If ϕ_1, ϕ_2 only depend on the projection of x onto subspaces V_1, V_2 respectively, and V_1, V_2 are orthogonal to each other, i.e $V_1V_2^\top = 0$, then $\mathcal{P}_{k_1+k_2}(\phi_1\phi_2) = \phi_1\phi_2$.

Lemma 7 follows directly from the fact that the d-dimensional Hermite basis is formed from taking products of the 1-dimensional Hermite basis elements.

In the above expansion, if there are two indices i_1, i_2 each with exponent 1, then we get a $\psi_{i_1}(x)\psi_{i_2}(x)\prod_{j\geqslant 3}\psi_{i_j}(x)^{z_j}$ term. By Lemma 7, this term is a polynomial with Hermite degree at least $2k$. Equivalently

$$\mathcal{P}_k\left(\psi_{i_1}(x)\psi_{i_2}(x)\prod_{j\geqslant 3}\psi_{i_j}(x)^{z_j}\right) = 0.$$

This is because $\psi_i(x)$ only depends on $v_{i,1}^\top x, \ldots, v_{i,J_i}^\top x$ and $\{v_{i,j}\}_{i\in[L],j\in[J_i]}$ are orthogonal vectors. Similarly, for terms of the form $\psi_{i_1}(x)\prod_{j\geqslant 2}\psi_{i_j}(x)^{z_j}$, we have that

$$\mathcal{P}_k\left(\psi_{i_1}(x)\prod_{j\geqslant 2}\psi_{i_j}(x)^{z_j}\right) = \psi_{i_1}(x)\mathcal{P}_0\left(\prod_{j\geqslant 2}\psi_{i_j}(x)^{z_j}\right).$$

Altogether, this gives (7) above.

Let us firstly compute the $\mathcal{P}_0$ terms in the above equation (7).

Case I. Firstly consider the case that w is odd and $w = 2s+1$. Then we have

$$\begin{aligned}\sum_{z_i\geqslant 2,q,z_1+\cdots+z_q=w-1,i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q} = \sum_{i_j\neq i}\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\\ + \sum_{z_i\geqslant 2,q<s,z_1+\cdots+z_q=w-1,i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\end{aligned}$$

For the first term, we have

$$\mathcal{P}_0\left(\sum_{i_j\neq i}\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\phi_{S_{i_s}}(x)^2\right) = \sum_{i_j\neq i}\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\,\mathbb{E}\left[\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right] = \frac{w!}{2^s}\sum_{i_j\neq i}\lambda_{i_1}^2\ldots\lambda_{i_s}^2 \tag{8}$$

For the second term, we count the number of monomials to get

$$\begin{aligned}
&\left|\mathcal{P}_0\left(\sum_{z_i\geqslant 2, q<s, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right| \\
&\qquad\leqslant \sum_{z_i\geqslant 2, q<s, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\left|\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\,\mathbb{E}\left[\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right]\right| \qquad (9)\\
&\qquad\lesssim \sum_{z_i\geqslant 2, q<s, z_1+\cdots+z_q=w-1, i_j\neq i}\left|\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\right| \\
&\qquad\lesssim L^{s-1}
\end{aligned}$$

In the second inequality, we use Gaussian hypercontractivity, Lemma 31.

Combining equation (8) and (9) together, and noticing that

$$\left|\sum_{i_j\neq i}\lambda_{i_1}^2\ldots\lambda_{i_s}^2-\sum_{i_j}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\right|\leqslant s\lambda_i^2\sum_{i_j\neq i}\lambda_{i_1}^2\ldots\lambda_{i_{s-1}}^2\lesssim L^{s-1}$$

which can help us substitute $\sum_{i_j}\lambda_{i_1}^2\ldots\lambda_{i_s}^2$ for $\sum_{i_j\neq i}\lambda_{i_1}^2\ldots\lambda_{i_s}^2$, we can have

$$\begin{aligned}
&\frac{1}{L^{w/2}}\left(\sum_{i=1}^{L}\lambda_i\psi_i(x)\mathcal{P}_0\left(\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right)\\
&\qquad=\frac{1}{L^{w/2}}\left(\frac{w!}{2^s}\sum_{i_j}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\right)\left(\sum_{i=1}^{L}\lambda_i\psi_i(x)\left(1+K_i\right)\right)
\end{aligned}$$

where $\sup_i|K_i|\lesssim 1/L$.

Case II. Secondly we will consider the case that w is even and denote $w=2s$. In that case, we observe that

$$\begin{aligned}
&\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\\
&\qquad=\sum_{z_i\geqslant 2, q<s, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}
\end{aligned}$$

By a similar argument like equation (9),

$$\sup_{1\leqslant i\leqslant L}\left|\sum_{z_i\geqslant 2, q<s, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\,\mathbb{E}\left[\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right]\right|\lesssim L^{s-1}$$

Therefore, we have the following bound for the $\mathcal{P}_0$ terms in our equation (7).

$$\begin{aligned}
&\frac{1}{L^{w/2}}\left(\sum_{i=1}^{L}\lambda_i\psi_i(x)\mathcal{P}_0\left(\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w-1, i_j\neq i}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right)\\
&\qquad=\sum_{i=1}^{L}\lambda_iK_i\psi_i(x)
\end{aligned}$$

where $\sup_i|K_i|\lesssim 1/L$.

Then let us compute the $\mathcal{P}_k$ terms. Firstly, we divide the monomials into two groups

$$\begin{aligned}
&\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\\
&=\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}+\sum_{2q=w, i_j}\frac{w!}{2^q}\lambda_{i_1}^2\ldots\lambda_{i_q}^2\psi_{i_1}(x)^2\ldots\psi_{i_q}(x)^2
\end{aligned}$$

For the first group, we have the following

$$\begin{aligned}
&\left\|\mathcal{P}_k\left(\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right\|_{L^2(\gamma)}^2\\
&\qquad\leqslant\left\|\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right\|_{L^2(\gamma)}^2\\
&\qquad\leqslant (wL)^{\lceil w/2\rceil-1}\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\left\|\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right\|_{L^2(\gamma)}^2\\
&\qquad\lesssim L^{2\lceil w/2\rceil-2}
\end{aligned}$$

In the second equality we use Gaussian hypercontractivity, Lemma 31.

For the second group, we have that

$$\begin{aligned}
&\left\|\mathcal{P}_k\left(\sum_{i_l}\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2=\left\|\sum_{i_l}\mathcal{P}_k\left(\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2\\
&\quad=\left(\frac{w!}{2^s}\right)^2\sum_{i_l}\sum_{j_l}\langle\mathcal{P}_k\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right),\mathcal{P}_k\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)\rangle_{L^2(\gamma)}\\
&\quad=\left(\frac{w!}{2^s}\right)^2\sum_{i_l,j_l,\{i_l\}\bigcap\{j_l\}\neq\varnothing}\langle\mathcal{P}_k\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right),\mathcal{P}_k\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)\rangle_{L^2(\gamma)}\\
&\quad\leqslant\left(\frac{w!}{2^s}\right)^2 s^2L^{2s-1}\sup_{i_l}\left\|\mathcal{P}_k\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2\\
&\quad\lesssim L^{w-1}
\end{aligned}$$

From the second line to the third line, we use the fact that if $\{i_l\}\bigcap\{j_l\}=\varnothing$, then $\mathcal{P}_k\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)$ and $\mathcal{P}_k\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)$ are two independent mean-zero random variables. Also, the third line to the fourth line is just counting the number of pairs of tuples with nonempty intersections. The fourth line to the fifth line is using gaussian hypercontractivity, Lemma 31, to bound the moments.

In a word, we have derived for any $k\geqslant 2$, and any $w\geqslant 2$ that

$$\left\|\frac{1}{L^{w/2}}\mathcal{P}_k\left(\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right\|_{L^2(\gamma)}=\mathcal{O}(L^{-1/2})$$

Sum up all the derivations above, and we get the following conclusion.

Lemma 8. *Given $k\geqslant 2$,*

- *When $w=2s+1$ with $s\geqslant 1$, we have*

$$\left\|\mathcal{P}_k(p(x)^w)-\frac{w!}{2^sL^s}\left(\sum_{i_j}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\right)p(x)\right\|_{L^2(\gamma)}=\mathcal{O}(L^{-1/2})$$

- *When $w=2s$ with $s\geqslant 1$, we have*

$$\|\mathcal{P}_k(p(x)^w)\|_{L^2(\gamma)}=\mathcal{O}(L^{-1/2})$$

Recall our $g(z) = \sum_{0 \leqslant i \leqslant q} g_i z^i$. After the projection, the feature that we get is approximately $\left(\sum_s \frac{1}{2^s L^s}(2s+1)! g_{2s+1} \left(\sum_{i_j} \lambda_{i_1}^2 \dots \lambda_{i_s}^2\right)\right) p$. Precisely speaking, we have

$$\left\| \mathcal{P}_k h - \left(\sum_s \frac{1}{2^s L^s} (2s+1)! c_{2s+1} \left(\sum_{i_j} \lambda_{i_1}^2 \dots \lambda_{i_s}^2 \right) \right) p \right\|_{L^2(\gamma)} = \mathcal{O}(L^{-1/2}) \tag{10}$$

Let's recall $\sum_i \lambda_i^2 = L$, so that informally speaking, we expect $p(x) \sim \mathcal{N}(0,1)$ in a limiting sense due to central limit theorem when L is large and λ_i are somehow balanced. Again, from the main text, it is tempting to conjecture some kind of approximated Stein's Lemma like

$$\mathcal{P}_k(g \circ p) \approx \mathbb{E}_{z \sim \mathcal{N}(0,1)} [g'(z)] \, p$$

Now we will verify this is indeed right. In our case, the derivative of g is $g'(z) = g_1 + 2g_2 z + 3g_3 z^2 + \dots + q g_q z^{q-1}$, and we can compute that $\mathbb{E}_{z \sim \mathcal{N}(0,1)} [g'(z)] = \sum_s g_{2s+1}(2s+1)!!$. Furthermore, we have

$$L^s = \left(\sum_i \lambda_i^2 \right)^s = \mathcal{O}(L^{s-1}) + s! \left(\sum_{i_j} \lambda_{i_1}^2 \dots \lambda_{i_s}^2 \right)$$

And as a direct consequence, we have

$$\frac{1}{2^s L^s}(2s+1)! g_{2s+1} \left(\sum_{i_j} \lambda_{i_1}^2 \dots \lambda_{i_s}^2 \right) = (2s+1)!! g_{2s+1} + \mathcal{O}(L^{-1})$$

Simply plugging the above equation in equation (10), we get our final result. □

Proof of Lemma 6. Firstly, we compute the hermite degree m components of $p(x)^w$, $w \geqslant 2$. From the definition of $\mathcal{P}_m$ and multinomial theorem, we know

$$\mathcal{P}_m(p(x)^w) = \frac{1}{L^{w/2}} \mathcal{P}_m \left(\sum_{z_i \geqslant 2, q, z_1 + \dots + z_q = w, i_j} \frac{w!}{z_1! \dots z_q!} \lambda_{i_1}^{z_1} \dots \lambda_{i_q}^{z_q} \psi_{i_1}(x)^{z_1} \dots \psi_{i_q}(x)^{z_q} \right)$$

by expanding $\left(\frac{1}{\sqrt{L}} \left(\sum_{1 \leqslant i \leqslant L} \lambda_i \psi_i(x)\right)\right)^w$ and computing the projection for each term. In the above expansion, if there is one index i_1 with exponent 1, then we get a $\psi_{i_1}(x) \prod_{j \geqslant 2} \psi_{i_j}(x)^{z_j}$ term. By Lemma 7, this term is a polynomial with Hermite degree at least k. As a result,

$$\mathcal{P}_m \left(\psi_{i_1}(x) \prod_{j \geqslant 2} \psi_{i_j}(x)^{z_j} \right) = 0.$$

This is because $\psi_i(x)$ only depends on $v_{i,1}^\top x, \dots, v_{i,J_i}^\top x$ and $\{v_{i,j}\}_{i \in [L], j \in [J_i]}$ are orthogonal vectors.

Firstly, notice that

$$\begin{aligned}
&\sum_{z_i \geqslant 2, q, z_1 + \dots + z_q = w, i_j} \frac{w!}{z_1! \dots z_q!} \lambda_{i_1}^{z_1} \dots \lambda_{i_q}^{z_q} \psi_{i_1}(x)^{z_1} \dots \psi_{i_q}(x)^{z_q} \\
&= \sum_{z_i \geqslant 2, 2q < w, z_1 + \dots + z_q = w, i_j} \frac{w!}{z_1! \dots z_q!} \lambda_{i_1}^{z_1} \dots \lambda_{i_q}^{z_q} \psi_{i_1}(x)^{z_1} \dots \psi_{i_q}(x)^{z_q} + \sum_{2q = w, i_j} \frac{w!}{2^q} \lambda_{i_1}^2 \dots \lambda_{i_q}^2 \psi_{i_1}(x)^2 \dots \psi_{i_q}(x)^2
\end{aligned}$$

For the first term, we have the following estimation

$$\begin{aligned}
&\left\|\mathcal{P}_m\left(\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right\|_{L^2(\gamma)}^2\\
&\qquad\leqslant\left\|\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right\|_{L^2(\gamma)}^2\\
&\qquad\lesssim d^{\lceil w/2\rceil-1}\sum_{z_i\geqslant 2, 2q<w, z_1+\cdots+z_q=w, i_j}\left\|\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right\|_{L^2(\gamma)}^2\\
&\qquad\lesssim d^{2\lceil w/2\rceil-2}
\end{aligned}$$

From the third line to the fourth line we use Gaussian hypercontractivity, Lemma 31 in Appendix E.2 to bound the high order moments of hermite polynomials. And for the second term, we only need to consider the case that $w=2s$ is even. In that case,

$$\begin{aligned}
&\left\|\mathcal{P}_m\left(\sum_{i_l}\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2=\left\|\sum_{i_l}\mathcal{P}_m\left(\frac{w!}{2^s}\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2\\
&=\left(\frac{w!}{2^s}\right)^2\sum_{i_l}\sum_{j_l}\langle\mathcal{P}_m\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right),\mathcal{P}_m\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)\rangle_{L^2(\gamma)}\\
&=\left(\frac{w!}{2^s}\right)^2\sum_{i_l,j_l,\{i_l\}\bigcap\{j_l\}\neq\varnothing}\langle\mathcal{P}_m\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right),\mathcal{P}_m\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)\rangle_{L^2(\gamma)}\\
&\lesssim\sup_{i_l}\left\|\mathcal{P}_m\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)\right\|_{L^2(\gamma)}^2 d^{2s-1}\\
&\lesssim d^{w-1}
\end{aligned}$$

From the second line to the third line, we use the fact that if $\{i_l\}\bigcap\{j_l\}=\varnothing$, then $\mathcal{P}_m\left(\lambda_{i_1}^2\ldots\lambda_{i_s}^2\psi_{i_1}(x)^2\ldots\psi_{i_s}(x)^2\right)$ and $\mathcal{P}_m\left(\lambda_{j_1}^2\ldots\lambda_{j_s}^2\psi_{j_1}(x)^2\ldots\psi_{j_s}(x)^2\right)$ are two independent mean-zero random variables. From the third line to the fourth line, we are just counting the number of pairs of tuples with nonempty intersection which is $\mathcal{O}(d^{2s-1})$.

In a word, we have derived that

$$\left\|\frac{1}{L^{w/2}}\mathcal{P}_m\left(\sum_{z_i\geqslant 2, q, z_1+\cdots+z_q=w, i_j}\frac{w!}{z_1!\ldots z_q!}\lambda_{i_1}^{z_1}\ldots\lambda_{i_q}^{z_q}\psi_{i_1}(x)^{z_1}\ldots\psi_{i_q}(x)^{z_q}\right)\right\|_{L^2(\gamma)}=\mathcal{O}(L^{-1/2})$$

Write $g(z)=\sum_{0\leqslant i\leqslant q}g_i z^i$ and sum over all the terms, and we get the desired result. □

B.2 Special Cases

Orthogonal Decomposable Tensors. Firstly, we will consider the case that $p(x):=\langle A, He_k(x)\rangle$ and A is an orthogonal decomposable tensor

$$A=\frac{1}{\sqrt{L}}\left(\sum_{i=1}^L\lambda_i v_i^{\otimes k}\right)$$

where $\langle v_i, v_j\rangle=\delta_{ij}$. Using identities for the Hermite polynomials (Appendix E.1), one can rewrite the feature as

$$p(x)=\frac{1}{\sqrt{L}}\left(\sum_{i=1}^L\lambda_i\langle v_i^{\otimes k}, He_k(x)\rangle\right)=\frac{1}{\sqrt{L}}\left(\sum_{i=1}^L\lambda_i h_k(v_i^\top x)\right)$$

This kind of feature satisfies Assumption 4 with $J_i=1$ for all i, if we further assume the regularity conditions $\sup_i|\lambda_i|=\mathcal{O}(1)$ and $\sum_i\lambda_i^2=L$.

Sum of Sparse Parities. Secondly, we will consider the case that

$$A = \frac{1}{\sqrt{L}} \left(\sum_{i=1}^{L} \lambda_i \cdot v_{i,1} \otimes \cdots \otimes v_{i,k} \right)$$

where $\langle v_{i_1,j_1}, v_{i_2,j_2} \rangle = \delta_{i_1 i_2} \delta_{j_1 j_2}$. In that case, our feature can be rewritten as

$$p(x) = \frac{1}{\sqrt{L}} \left(\sum_{i=1}^{L} \lambda_i \langle v_{i,1} \otimes \cdots \otimes v_{i,k}, He_k(x) \rangle \right) = \frac{1}{\sqrt{L}} \left(\sum_{i=1}^{L} \lambda_i \left(\prod_{j=1}^{k} \langle v_{i,j}, x \rangle \right) \right)$$

This kind of feature also satisfies Assumption 4 with $J_i = k$ for all i, if we further assume the regularity conditions $\sup_i |\lambda_i| = \mathcal{O}(1)$ and $\sum_i \lambda_i^2 = L$.

For a concrete example, when $v_{i,j} = e_{k(i-1)+j}$ and $L = d/k$,

$$p(x) = \frac{1}{\sqrt{d/k}} \left(\lambda_1 x_1 x_2 \dots x_k + \cdots + \lambda_{d/k} x_{d-k+1} \dots x_d \right)$$

and hence the name "sum of sparse parities".

C Proof of Lemma 1

The goal in this appendix is to prove Lemma 1, which is restated below:

Lemma 1. *For any constant $\alpha \in (0,1)$, any $m_1 \geqslant d^{k+\alpha}$ and any $n \geqslant d^{k+3\alpha}$, set $T_1 = \mathrm{poly}(n, m_1, d)$, $\eta_1 = \frac{1}{\mathrm{poly}(n,m_1,d)}$ and $\xi_1 = \frac{2m_1}{d^{k+\alpha}}$. Then, for any absolute constant $\delta \in (0,1)$, with probability at least $1 - \delta/2$ over the initialization V, s and training data $\mathcal{D}_1$, we have*

$$\| h_{\theta^{(T_1)}} - \mathcal{P}_{\leqslant k} h \|_{L^2(\gamma)}^2 = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

Proof Outline. Throughout the first stage of Algorithm 1, c remains at 0. Consequently, during this stage, the network is given by

$$g_{u,s,V}(x) = u^\top \sigma_1(Vx + s)$$

where σ_1 is a degree k polynomial. Given that V, s is kept constant and only u is trained, the network is equivalent to a random feature model with the random feature $\sigma_1(Vx + s)$.

The proof proceeds in three steps:

- First, we show that there exists u^* such that $g_{u^*,s,V}$ approximates $\mathcal{P}_k h$, the degree k component of the target.
- Next, we leverage strong convexity of the empirical loss minimization problem to show that GD can find an approximate global minimizer in polynomial time.
- Finally, we invoke a kernel Rademacher complexity argument to bound the test performance.

In this section, we may use $\sigma(\cdot)$ to refer $\sigma_1(\cdot)$, and m to refer m_1 due to notation simplicity.

C.1 Approximation

First, we show that when σ is a k degree polynomial, the random feature model can and only can approximate the degree $\leqslant k$ part of the target function.

Lemma 9. *For any $u \in \mathbb{R}^m$, we have the following equality for any function $h \in L^2(\mathbb{R}^d, \gamma)$*

$$\| g_{u,s,V} - h \|_{L^2(\gamma)}^2 = \| g_{u,s,V} - \mathcal{P}_{\leqslant k} h \|_{L^2(\gamma)}^2 + \| \mathcal{P}_{\leqslant k} h - h \|_{L^2(\gamma)}^2$$

Remark 4. From Lemma 9, we can see when we try to approximate h using $g_{u,s,V}$, we are actually trying our best to approximate $\mathcal{P}_{\leqslant k} h$. That is to say,

$$\operatorname*{argmin}_u \| g_{u,s,V} - h \|_{L_2(\gamma)}^2 = \operatorname*{argmin}_u \| g_{u,s,V} - \mathcal{P}_{\leqslant k} h \|_{L_2(\gamma)}^2$$

Proof. By a direct computation, we have

$$\begin{aligned}\|g_{u,s,V}-h\|_{L^2(\gamma)}^2 &= \left\|u^\top\sigma(Vx+s)-\sum_j\langle H_j,He_j(x)\rangle\right\|_{L^2(\gamma)}^2\\ &= \left\|u^\top\sigma(Vx+s)-\sum_{j\leqslant k}\langle H_j,He_j(x)\rangle\right\|_{L^2(\gamma)}^2+\left\|\sum_{j\geqslant k+1}\langle H_j,He_j(x)\rangle\right\|_{L^2(\gamma)}^2\\ &= \|g_{u,s,V}-\mathcal{P}_{\leqslant k}h\|_{L^2(\gamma)}^2+\|h-\mathcal{P}_{\leqslant k}h\|_{L^2(\gamma)}^2\end{aligned} \tag{11}$$

where $H_j = \mathbb{E}_x[h(x)He_j(x)]$. Here we use the hermite expansion which we state in Appendix E.1. □

We next show that $\mathcal{P}_k h$ can be expressed by an infinite-width network by the following three lemmas.

Lemma 10. *There exists $f:\mathbb{S}^{d-1}\to\mathbb{R}$ such that*

$$\mathbb{E}_v[f(v)h_k(v^\top x)] = (\mathcal{P}_k h)(x) \quad \textit{and} \quad \mathbb{E}_v[f(v)^2] = \mathcal{O}(d^k).$$

where v obeys the uniform distribution on $\mathbb{S}^{d-1}$.

Proof. Recall that $(\mathcal{P}_k h)(x)$ can be represented as $\langle A, He_k(x)\rangle$ for some symmetric tensor $A\in(\mathbb{R}^d)^{\otimes k}$. Furthermore, observing that

$$\mathbb{E}_v\left[f(v)h_k(v^\top x)\right] = \langle \mathbb{E}_v\left[f(v)v^{\otimes k}\right], He_k(x)\rangle$$

by Lemma 28, it suffices to solve for $u(\cdot)$ such that $\mathbb{E}_v[f(v)v^{\otimes k}] = A$.

Let $\mathrm{Vec}:(\mathbb{R}^d)^{\otimes k}\to\mathbb{R}^{d^k}$ be the unfolding operator. We claim that one solution for f is

$$f(v) = \mathrm{Vec}(v^{\otimes k})^\top\left(\mathbb{E}_v\,\mathrm{Vec}(v^{\otimes k})\,\mathrm{Vec}(v^{\otimes k})^\top\right)^\dagger\mathrm{Vec}(A).$$

First, by Corollary 42 in Damian et al. (2022), we have

$$\mathbb{E}_{x\sim\gamma}\left[\mathrm{Vec}(x^{\otimes k})\,\mathrm{Vec}(x^{\otimes k})^\top\right] \succeq k!\Pi_{\mathrm{Sym}^k(\mathbb{R}^d)}, \tag{12}$$

where $\Pi_{\mathrm{Sym}^k(\mathbb{R}^d)}$ is the projection operator onto symmetric k tensors. Since A is symmetric, we indeed see that

$$\mathrm{Vec}\left(\mathbb{E}_v\left[f(v)v^{\otimes k}\right]\right) = \mathbb{E}_v\,\mathrm{Vec}(v^{\otimes k})\,\mathrm{Vec}(v^{\otimes k})^\top\left(\mathbb{E}_v\,\mathrm{Vec}(v^{\otimes k})\,\mathrm{Vec}(v^{\otimes k})^\top\right)^\dagger\mathrm{Vec}(A) = \mathrm{Vec}(A).$$

Plugging this back to $\mathbb{E}_v\left[f(v)^2\right]$ and applying the Cauchy inequality, we get

$$\mathbb{E}_v\left[f(v)^2\right] \leqslant \lambda_{\max}\left(\left(\mathbb{E}_v\,\mathrm{Vec}(v^{\otimes k})\,\mathrm{Vec}(v^{\otimes k})^\top\right)^\dagger\right)\|\,\mathrm{Vec}(A)\|^2 \tag{13}$$

Therefore, to estimate the L^2 norm of $f(v)$ we only need to look at the spectrum of the matrix above.

For $X\sim\mathcal{N}(0,I_d)$, it is clear that YZ shares the same distribution with X, where $Y\sim\chi(d)$ and $Z\sim\mathrm{Unif}(\mathbb{S}^{d-1})$ and Y,Z are independent. Therefore,

$$\mathbb{E}_X\left[\mathrm{Vec}(X^{\otimes k})\,\mathrm{Vec}(X^{\otimes k})^\top\right] = \mathbb{E}_Y\left[Y^{2k}\right]\mathbb{E}_Z\left[\mathrm{Vec}(Z^{\otimes k})\,\mathrm{Vec}(Z^{\otimes k})^\top\right] \leqslant d^k\,\mathbb{E}_Z\left[\mathrm{Vec}(Z^{\otimes k})\,\mathrm{Vec}(Z^{\otimes k})^\top\right]$$

due to Lemma 44 in Damian et al. (2022). Furthermore, we get $\lambda_{\max}\left(\left(\mathbb{E}_X\left[\mathrm{Vec}(X^{\otimes k})\,\mathrm{Vec}(X^{\otimes k})^\top\right]\right)^\dagger\right) \leqslant \frac{1}{k!}$ by equation (12). Plugging this back to equation (13), we will have

$$\mathbb{E}_v\left[f(v)^2\right] \leqslant \frac{1}{k!}d^k\|\,\mathrm{Vec}(A)\|^2 \lesssim d^k,$$

where we used the fact that $\|\,\mathrm{Vec}(A)\|_2^2 = \|A\|_F^2 = \mathbb{E}\left[(\mathcal{P}_k h)(x)^2\right] = \mathcal{O}(1)$. □

Lemma 11. *Let $s \sim \mathcal{N}(0,1)$. Then, there exists $w : \mathbb{R} \to \mathbb{R}$ with $\mathbb{E}_s[w(s)^2] = \mathcal{O}(1)$ and*

$$\mathbb{E}_s\left[w(s)\sigma\left(\frac{z+s}{\sqrt{2}}\right)\right] = h_k(z).$$

Proof. One has the following Hermite addition formula:

$$h_i\left(\frac{z+s}{\sqrt{2}}\right) = 2^{-i/2}\sum_{j=0}^{i}\binom{i}{j}^{1/2} h_{i-j}(s)h_j(z).$$

Thus writing $\sigma(z) = \sum_{i\geqslant 0} c_i h_i(z)$, we have

$$\begin{aligned}\sigma\left(\frac{z+s}{\sqrt{2}}\right) &= \sum_{i\geqslant 0}\sum_{j=0}^{i} c_i 2^{-i/2}\binom{i}{j}^{1/2} h_{i-j}(s)h_j(z)\\ &= \sum_{j\geqslant 0} h_j(z)\sum_{i=j}^{k} c_i 2^{-i/2}\binom{i}{j}^{1/2} h_{i-j}(s).\end{aligned}$$

Define $w_0, \ldots, w_k$ recursively by

$$\begin{aligned}w_0 &= c_k^{-1}2^{k/2}\\ w_j &= -c_k^{-1}2^{k/2}\binom{k}{j}^{-1/2}\left(\sum_{i=0}^{j-1} c_{k+i-j}2^{-(k+i-j)/2}\binom{k+i-j}{i}^{1/2} w_i\right).\end{aligned}$$

As a consequence, for $j \geqslant 1$, we have

$$0 = \sum_{i=0}^{j} c_{k+i-j}2^{-(k+i-j)/2}\binom{k+i-j}{i}^{1/2} w_i.$$

Therefore for all $0 \leq j \leq k-1$, we have

$$\begin{aligned}0 &= \sum_{i=0}^{k-j} c_{i+j}2^{-(i+j)/2}\binom{i+j}{i}^{1/2} w_i\\ &= \sum_{i=j}^{k} c_i 2^{-i/2}\binom{i}{j}^{1/2} w_{i-j}.\end{aligned}$$

Setting $w(s) = \sum_{i=0}^{k} w_i h_i(s)$, we thus have that

$$\begin{aligned}\mathbb{E}_s\left[w(s)\sigma\left(\frac{z+s}{\sqrt{2}}\right)\right] &= \sum_{j\geqslant 0} h_j(z)\sum_{i=j}^{k} c_i 2^{-i/2}\binom{i}{j}^{1/2} w_{i-j}\\ &= 2^{-k/2}c_k w_0 h_k(z) + \sum_{j\geqslant 0}^{k-1} h_j(z)\left(\sum_{i=j}^{k} c_i 2^{-i/2}\binom{i}{j}^{1/2} w_{i-j}\right)\\ &= h_k(z),\end{aligned}$$

as desired. Since we regard k as a constant, and we have $\sup_i |c_i| = \mathcal{O}(1)$ and $c_k = \Theta(1)$ due to Assumption 6, the norm bound follows. □

Lemma 12. *There exists $u : \mathbb{S}^{d-1} \times \mathbb{R} \to \mathbb{R}$ such that*

$$\mathbb{E}_{v,s}\left[u(v,s)\sigma\left(\frac{v^\top x + s}{\sqrt{2}}\right)\right] = (\mathcal{P}_k h)(x) \text{ and } \mathbb{E}_{v,s}\left[u(v,s)^2\right] = \mathcal{O}(d^k)$$

Proof. By Lemma 11, we get $\mathbb{E}_s\left[w(s)\sigma\left(\frac{z+s}{\sqrt{2}}\right)\right] = h_k(z)$ for some $\mathbb{E}_s\left[w(s)^2\right] = \mathcal{O}(1)$ and $w(\cdot)$ is a k degree polynomial. Substitute z with $v^\top x$, and then use Lemma 10, we have

$$\mathbb{E}_{v,s}\left[f(v)w(s)\sigma\left(\frac{v^\top x+s}{\sqrt{2}}\right)\right] = \mathbb{E}_v\left[f(v)h_k(v^\top x)\right] = (\mathcal{P}_k h)(x)$$

Set $u(v,s) = f(v)w(s)$. We next bound the L^2 norm of $u(v,s)$ by the independence between v and s.

$$\mathbb{E}\left[u(v,s)^2\right] = \mathbb{E}\left[f(v)^2 w(s)^2\right] = \mathbb{E}\left[f(v)^2\right]\mathbb{E}\left[w(s)^2\right] \lesssim d^k$$

□

Remark 5. In the above lemma, our feature is $\sigma\left(\frac{v^T x+s}{\sqrt{2}}\right)$ with v uniformly sampled from the unit sphere and s sampled from $\mathcal{N}(0,1)$. This is equivalent with our feature $\sigma(v^T x+s)$ in the main text, with v uniformly sampled from the sphere of radius $\frac{1}{\sqrt{2}}$ and s sampled from $\mathcal{N}(0,1/2)$. We will use the $\sigma\left(\frac{v^T x+s}{\sqrt{2}}\right)$ formulation in the remainder of the section without loss of generality.

Next, we show that we can use this infinite width construction to construct a finite-width network that approximates $\mathcal{P}_k h$.

Lemma 13. *For any absolute constant $\delta \in (0,1)$ and $m \in \mathbb{N}^+$, with probability at least $1-\delta/8$ over the sampling of V, s, there exists u^* such that*

$$\|g_{u^*,s,V} - \mathcal{P}_k h\|^2_{L^2(\gamma)} = \mathcal{O}(m^{-1}d^k) \text{ and } \|u^*\|^2 = \mathcal{O}(m^{-1}d^k)$$

Remark 6. Due to Lemma 2 and utilizing the lemma above, we have

$$\|g_{u^*,s,V} - \mathcal{P}_{\leqslant k} h\|^2_{L^2(\gamma)} \lesssim d^{-1} + m^{-1}d^k$$

Proof of Lemma 13. We use Monte Carlo sampling to help us construct the u^*. Let $u(\cdot,\cdot)$ be the function from Lemma 12, so that $(\mathcal{P}_k h)(x) = \mathbb{E}_{v,s}\left[u(v,s)\sigma\left(\frac{v^\top x+s}{\sqrt{2}}\right)\right]$. We sample $\Theta = \{v_i, s_i\}_{i=1}^m$ i.i.d. and set $u_i^* := \frac{1}{m}u(v_i,s_i)$. As such, one has that

$$\begin{aligned}
\mathbb{E}_\Theta\mathbb{E}_x\left|g_{u^*,s,V}(x) - (\mathcal{P}_k h)(x)\right|^2 &= \mathbb{E}_x\mathbb{E}_\Theta\left|\frac{1}{m}\sum_{j=1}^m u(v_j,s_j)\sigma\left(\frac{v_j^\top x+s_j}{\sqrt{2}}\right) - (\mathcal{P}_k h)(x)\right|^2 \\
&= \frac{1}{m^2}\mathbb{E}_x\sum_{j,l=1}^m \mathbb{E}_\Theta\left[\left(u(v_j,s_j)\sigma\left(\frac{v_j^\top x+s_j}{\sqrt{2}}\right)\right)\left(u(v_l,s_l)\sigma\left(\frac{v_l^\top x+s_l}{\sqrt{2}}\right)\right)\right] \\
&= \frac{1}{m^2}\sum_{j=1}^m \mathbb{E}_x\mathbb{E}_{v_j,s_j}\left[\left(u(v_j,s_j)\sigma\left(\frac{v_j^\top x+s_j}{\sqrt{2}}\right)\right)^2\right] \\
&\lesssim \frac{1}{m}\mathbb{E}_{v,s}\left[f(v)^2 w(s)^2(1+s^{2k})\right] \\
&\lesssim \frac{1}{m}\mathbb{E}_{v,s}\left[u(v,s)^2\right]
\end{aligned} \tag{14}$$

and

$$\mathbb{E}_\Theta\left[\frac{1}{m}\sum_{j=1}^m u(v_j,s_j)^2\right] = \mathbb{E}_{v,s}\left[u(v,s)^2\right]$$

Therefore, from Markov inequality, we can derive that for any constant $K > 0$ we have

$$\mathbb{P}_\Theta\left(\mathbb{E}\left|g_{u^*,s,V} - \mathcal{P}_k h\right|^2 \geqslant \Theta(1)\frac{K}{m}\mathbb{E}\left[u(v,s)^2\right]\right) \leqslant \frac{1}{K} \tag{15}$$

and

$$\mathbb{P}_\Theta\left(\frac{1}{m}\sum_{j=1}^{m} u(v_j,s_j)^2 \geqslant K\,\mathbb{E}\left[u(v,s)^2\right]\right) \leqslant \frac{1}{K}$$

for some $\Theta(1)$. Setting $1/K = \delta/16$, plugging in the bound on $\mathbb{E}\left[u(v,s)^2\right]$ from Lemma 12 and noting that $\|u^*\|^2 = \frac{1}{m^2}\sum_{i=1}^{m} u(v_j,s_j)^2$ yields the desired result. □

Throughout the remainder of this section, we let $\epsilon_1 = \Theta(1)\frac{K}{m}\mathbb{E}\left[u(v,s)^2\right]$ for notation simplicity where the $\Theta(1)$ is from equation (15). Since we see δ, K as absolute constants, we have $\epsilon_1 = \mathcal{O}(d^k/m)$.

C.2 Empirical Performance

Next, we focus on the concentration over the population loss given by

$$L(u) = \|g_{u,s,V} - h\|^2_{L^2(\gamma)}$$

evaluated at the point $u = u^*$, which is defined in our Lemma 13. Our primary tool for this concentration is Corollary 3. For the sake of notational clarity, let us define $\hat{L}(u) := \frac{1}{n}\sum_{i=1}^{n}\left(g_{u,s,V}(x_i) - h(x_i)\right)^2$ to represent the empirical loss based on the initial dataset $\mathcal{D}_1$.

Lemma 14. *Under the setup and the results in Lemma 13, we will have with probability at least* $1-\delta/4$,

$$\left|\hat{L}(u^*) - L(u^*)\right| \lesssim \frac{1}{\sqrt{n}}$$

Proof. By Corollary 3, for any $\beta > 0$, we have

$$\mathbb{P}\left[\left|\hat{L}(u^*) - L(u^*)\right| \geqslant \beta\frac{1}{\sqrt{n}}\sqrt{\mathrm{Var}\left((g_{u^*,s,V}-h)^2\right)}\right] \leqslant 2\exp\left(-\Theta(1)\min(\beta^2,\beta^{1/r})\right)$$

Moreover,

$$\begin{aligned}\mathrm{Var}\left((g_{u^*,s,V}-h)^2\right) \leqslant \mathbb{E}_x\left[(g_{u^*,s,V}(x)-h(x))^4\right] &\lesssim \left(\mathbb{E}_x\left[(g_{u^*,s,V}(x)-h(x))^2\right]\right)^2 \\ &\lesssim \left(\epsilon_1 + \mathbb{E}_x\left[h(x)^2\right]\right)^2 \lesssim 1,\end{aligned}$$

where the second inequality relies on Gaussian hypercontractivity (Lemma 31), and the final step sets $m \geqslant d^{k+\alpha}$ so that $\epsilon_1 \lesssim 1$. Plugging this back and choosing some $\beta = \Theta(1)$ finishes the proof. □

Observe that during the first stage of Algorithm 1, we are solving the following minimization problem:

$$\min_u \hat{L}(u) + \frac{1}{2}\xi_1\|u\|^2 \tag{16}$$

Since this problem is strongly convex and smooth, plain GD can converge to an approximate minimizer exponentially fast. The next lemma bounds the time needed to obtain a small empirical loss:

Lemma 15. *Set* $\xi_1 = \frac{2m}{d^{k+\alpha}}$. *For any* $\epsilon_2 \in (0,1)$, *let* $T_1 \gtrsim m(\log m)^k\log(m/\epsilon_2)$. *Then, when* m, n *are larger than some absolute constant, with probability at least* $1-3\delta/8$, *the predictor* $\hat{u} := u^{(T_1)}$ *satisfies*

$$\hat{L}(\hat{u}) \leqslant \epsilon_1 + \|h - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} + \mathcal{O}(d^{-\alpha}) + \mathcal{O}(1)\frac{1}{\sqrt{n}} + \epsilon_2$$

and $\|\hat{u}\|^2 \lesssim \frac{d^{k+\alpha}}{m}$.

Proof. If $\hat{u}$ is an ϵ_2-minimizer of (16), then we have

$$\hat{L}(\hat{u}) + \frac{1}{2}\xi_1 \|\hat{u}\|^2 \leqslant \hat{L}(u^*) + \frac{1}{2}\xi_1 \|u^*\|^2 + \epsilon_2 \leqslant L(u^*) + \frac{1}{2}\xi_1 \|u^*\|^2 + \mathcal{O}(1)\frac{1}{\sqrt{n}} + \epsilon_2$$

By choosing $\xi_1 = \frac{2m}{d^{k+\alpha}}$, we get

$$\frac{m}{d^{k+\alpha}} \|\hat{u}\|^2 \lesssim \epsilon_1 + d^{-\alpha} + \|h - \mathcal{P}_{\leqslant k} h\|^2_{L^2(\gamma)} + \frac{1}{\sqrt{n}} + \epsilon_2 \lesssim 1$$

At the same time, we will also have

$$\hat{L}(\hat{u}) \leqslant \epsilon_1 + \mathcal{O}(d^{-\alpha}) + \|h - \mathcal{P}_{\leqslant k} h\|^2_{L^2(\gamma)} + \mathcal{O}(1)\frac{1}{\sqrt{n}} + \epsilon_2$$

It thus suffices to analyze the optimization problem (16).

Clearly, this convex optimization problem is at least 2-strongly convex. To estimate the time complexity, we also need to estimate the smoothness of our optimization objective.

Lemma 16. *With probability at least* $1 - \mathcal{O}(1/m)$,

$$\left\|\nabla \hat{L}(u_1) - \nabla \hat{L}(u_2)\right\| \lesssim m(\log m)^k \|u_1 - u_2\|$$

Proof. We calculate the gradient out

$$\nabla \hat{L}(u) = \frac{1}{n}\sum_{i=1}^n 2\left(u^\top \sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right) - h(x_i)\right)\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)$$

and then bound the Lipschitz constant of the gradient

$$\begin{aligned}\left\|\nabla \hat{L}(u_1) - \nabla \hat{L}(u_2)\right\| &= \left\|\frac{2}{n}\sum_{i=1}^n \langle u_1 - u_2, \sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\rangle \sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\| \\ &\leqslant \left(\frac{2}{n}\sum_{i=1}^n \left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\|^2\right)\|u_1 - u_2\|\end{aligned}$$

Using Corollary 3, we have the following concentration inequality for any $\beta \geqslant 1$

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \sigma\left(\frac{v_j^\top x_i + s_j}{\sqrt{2}}\right)^2 - \mathbb{E}_x\left[\sigma\left(\frac{v_j^\top x + s_j}{\sqrt{2}}\right)^2\right]\right| \geqslant \beta\frac{1}{\sqrt{n}}\sqrt{\mathrm{Var}\left(\sigma\left(\frac{v_j^\top x + s_j}{\sqrt{2}}\right)^2\right)}\right) \leqslant 2e^{-\Theta(1)\beta^{1/k}}$$

Furthermore, estimating $\mathbb{E}_x\left[\sigma\left(\frac{v_j^\top x + s_j}{\sqrt{2}}\right)^2\right]$, $\mathrm{Var}\left(\sigma\left(\frac{v_j^\top x + s_j}{\sqrt{2}}\right)^2\right)$ and doing union bound over all v_j, we get the following inequality with probability at least $1 - 2me^{-\Theta(1)\beta^{1/k}}$

$$\frac{1}{n}\sum_{i=1}^n \left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\|^2 \lesssim \left(1 + \beta\frac{1}{\sqrt{n}}\right)\sum_{j=1}^m (1 + s_j^{2k})$$

By Corollary 3 again, we can concentrate $\frac{1}{m}\sum_{j=1}^m (1 + s_j^{2k})$ and get the following with probability at least $1 - 2e^{-\Theta(1)m^{1/2k}}$

$$\frac{1}{m}\sum_{j=1}^m (1 + s_j^{2k}) \lesssim 1$$

In that case, we choose $\beta = \Theta(1)(\log m)^k$ for some large $\Theta(1)$ and the lemma is proved. □

Having derived the above Lemma, using Lemma 36 in Appendix E.5, we can choose the learning rate $\eta_1 = \frac{1}{m(\log m)^k \Theta(1)}$ and have

$$\left\|u^{(t)} - u_{opt}\right\|^2 \leqslant \left(1 - \frac{1}{\Theta(1) m (\log m)^k}\right)^t \|u_{opt}\|^2$$

where u_{opt} is the unique optimal solution for that optimization problem.

In addition, in order to bound the empirical performance, we also need to upper bound the gradient.

$$\begin{aligned}
\sup_{\|u\| \leqslant R} \left\|\nabla \hat{L}(u) + 2u\right\| &\leqslant 2R + \frac{2}{n}\sum_{i=1}^n \left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\| \left|u^\top \sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right) - h(x_i)\right| \\
&\leqslant 2R + \frac{2}{n}\sum_{i=1}^n \left(\left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\| h(x_i) + \left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\|^2 \|u\|\right) \\
&\leqslant 2R + \frac{2}{n}\sum_{i=1}^n \left((1+R)\left\|\sigma\left(\frac{Vx_i + s}{\sqrt{2}}\right)\right\|^2 + h(x_i)^2\right) \\
&\lesssim (1+3R) m (\log m)^k + \frac{2}{n}\sum_{i=1}^n h(x_i)^2
\end{aligned}$$

with probability at least $1 - \mathcal{O}(1/m)$. In order to bound $\frac{1}{n}\sum_i h(x_i)^2$, by Corollary 3, we have the following for any $\beta \geqslant 1$

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n h(x_i)^2 - \mathbb{E}_x\, h(x)^2\right| \geqslant \beta \frac{1}{\sqrt{n}} \sqrt{\mathrm{Var}(h(x)^2)}\right) \leqslant 2e^{-\Theta(1)\beta^{1/r}}$$

Therefore, by choosing $\beta = \Theta(1)(\log n)^r$ with some large $\Theta(1)$, with probability at least $1 - 1/n$, we have $\frac{1}{n}\sum_{i=1}^n h(x_i)^2 \lesssim 1$. In that case, we have

$$\hat{L}(u^{(t)}) + \left\|u^{(t)}\right\|^2 \leqslant \hat{L}(u_{opt}) + \|u_{opt}\|^2 + \sup_{\|u\| \leqslant 2\|u_{opt}\|} \left\|\nabla \hat{L}(u) + 2u\right\| \left\|u^{(t)} - u_{opt}\right\|$$

Since $\|u_{opt}\| = \mathcal{O}(1)$, $\sup_{\|u\| \leqslant 2\|u_{opt}\|} \left\|\nabla \hat{L}(u) + 2u\right\| = \mathcal{O}(m(\log m)^k)$, if we want

$$\sup_{\|u\| \leqslant 2\|u_{opt}\|} \left\|\nabla \hat{L}(u) + 2u\right\| \left\|u^{(t)} - u_{opt}\right\| \leqslant \epsilon_2$$

it is sufficient to have $T_1 \gtrsim m(\log m)^k \log(m/\epsilon_2)$.

□

C.3 Uniform Generalization Bounds

To conclude, we need to do a union bound over u for our population loss $\|g_{u,s,V} - h\|^2_{L^2(\gamma)}$. We first consider a truncated version of population loss, which allows us to invoke standard Rademacher complexity generalization bounds. We conclude by properly handling the truncation.

Proof of Lemma 1. Let us denote $\ell_\tau(x, y) = (x - y)^2 \wedge \tau^2$. Via standard Rademacher complexity generalization bounds, detailed in Lemmas 33, 34 and 35, recall that we see δ as an absolute constant, when m, n, d are larger than some absolute constant, we have that with probability at least $1 - \delta/16$

$$\begin{aligned}
\sup_{\|u\| \leqslant M_u} \left|\frac{1}{n}\sum_{i=1}^n \ell_\tau(g_{u,s,V}(x_i), h(x_i)) - \mathbb{E}_x\left[\ell_\tau(g_{u,s,V}(x), h(x))\right]\right| &\lesssim 2\,\mathrm{Rad}_n(\mathcal{F}) + \tau^2\sqrt{\frac{1}{n}} \\
&\leqslant 4\tau\,\mathrm{Rad}_n(\mathcal{G}) + \tau^2\sqrt{\frac{1}{n}} \\
&\lesssim 4\tau M_u \sqrt{\frac{m}{n}} + \tau^2\sqrt{\frac{1}{n}}
\end{aligned}$$

where $\mathcal{G} = \{g_{u,s,V} : \|u\| \leqslant M_u\}$ and $\mathcal{F} = \{\ell_\tau(g_{u,s,V}(\cdot), h(\cdot)) : \|u\| \leqslant M_u\}$. The first step is just standard uniform generalization bounds for bounded function class. The second step is via contraction lemma to compute the Rademacher complexity, and the third step is a direct calculation. So, by that bound, we can see $\mathbb{E}_x\left[\ell_\tau(g_{\hat{u},s,V}(x), h(x))\right]$ is well controlled for moderate large τ. Combining this with Lemma 15, with probability $1 - 7\delta/16$, we have

$$\mathbb{E}_x\left[\ell_\tau(g_{\hat{u},s,V}(x), h(x))\right] - \|h - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} \lesssim \tau M_u\sqrt{\frac{m}{n}} + \tau^2\sqrt{\frac{1}{n}} + \epsilon_1 + d^{-\alpha} + \frac{1}{\sqrt{n}} + \epsilon_2$$

Dealing with the Truncation. Based on the above arguments, to bound the L^2 generalization error, it suffices to control the quantity

$$\mathbb{E}_x\left[\left((g_{\hat{u},s,V}(x) - h(x))^2\right)\mathbf{1}_{|g_{\hat{u},s,V}(x)-h(x)|\geqslant\tau}\right]$$

This is done in the following lemma, whose proof is deferred to Appendix C.3.1

Lemma 17. *With probability at least $1 - \delta/32$, for any $\tau \gtrsim 1$, we have*

$$\mathbb{E}_x\left[\left((g_{\hat{u},s,V}(x) - h(x))^2\right)\mathbf{1}_{|g_{\hat{u},s,V}(x)-h(x)|\geqslant\tau}\right] \lesssim e^{-\Theta(1)\tau^{2/r}}$$

Altogether, when m, n, d are larger than some absolute constant, with probability at least $1 - \delta/2$, we have the following inequality

$$\begin{aligned}
&\|g_{\hat{u},s,V} - h\|^2_{L^2(\gamma)} - \|h - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} \\
&\quad\leqslant \mathbb{E}_x\left[\ell_\tau(g_{\hat{u},s,V}(x), h(x))\right] - \|h - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} + \mathbb{E}_x\left[\left((g_{\hat{u},s,V}(x) - h(x))^2\right)\mathbf{1}_{|g_{\hat{u},s,V}(x)-h(x)|\geqslant\tau}\right] \\
&\quad\lesssim \tau M_u\sqrt{\frac{m}{n}} + \tau^2\sqrt{\frac{1}{n}} + \epsilon_1 + d^{-\alpha} + \frac{1}{\sqrt{n}} + \epsilon_2 + \exp\left(-\Theta(1)\tau^{2/r}\right)
\end{aligned}$$

where we recall $\epsilon_1 = \mathcal{O}(m^{-1}d^k)$.

For any $\alpha \in (0, 1)$, select $\epsilon_2 = d^{-\alpha}$. Clearly we have $T_1 = \mathrm{poly}(n, m, d)$ and $\eta_1 = \frac{1}{\mathrm{poly}(n,m,d)}$ in that case. Recall that we have chosen the width $m \geqslant d^{k+\alpha}$, the sample size $n \geqslant d^{k+3\alpha}$, and we choose the truncation level to be $\tau = \Theta(1)(\log d)^{r/2}$ and $M_u^2 = \Theta\left(\frac{d^{k+\alpha}}{m}\right)$. Plugging those in yields

$$\begin{aligned}
\|g_{\hat{u},s,V} - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} &\leqslant \|g_{\hat{u},s,V} - h\|^2_{L^2(\gamma)} - \|h - \mathcal{P}_{\leqslant k}h\|^2_{L^2(\gamma)} + \mathcal{O}(1/d) \\
&\lesssim (\log d)^{r/2}d^{-\alpha} + (\log d)^r d^{-k/2-3\alpha/2} \\
&= \widetilde{\mathcal{O}}(d^{-\alpha}),
\end{aligned}$$

as desired. □

C.3.1 Proof of Lemma 17

Proof of Lemma 17. We will first use Cauchy inequality, then estimate the moments.

$$\begin{aligned}
\left(\mathbb{E}_x\left[\left((g_{\hat{u},s,V}(x) - h(x))^2\right)\mathbf{1}_{|g_{\hat{u},s,V}(x)-h(x)|\geqslant\tau}\right]\right)^2 &\leqslant \mathbb{E}_x\left[(g_{\hat{u},s,V}(x) - h(x))^4\right]\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau\right) \\
&\lesssim \left(\mathbb{E}_x\left[g_{\hat{u},s,V}(x)^4\right] + \mathbb{E}_x\left[h(x)^4\right]\right)\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau\right) \\
&\lesssim \left(\mathbb{E}_x\left[g_{\hat{u},s,V}(x)^2\right]^2 + \mathbb{E}_x\left[h(x)^2\right]^2\right)\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau\right)
\end{aligned} \tag{17}$$

The last step is by Gaussian hypercontractivity, Lemma 31. Recall $g_{u,s,V}(x) = u^\top\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)$. Notice that

$$\mathbb{E}_x\left[g_{u,s,V}(x)^2\right] = u^\top\,\mathbb{E}_x\left[\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)^\top\right]u \tag{18}$$

Therefore, we just need to give a tight bound for $\hat{u}^\top \mathbb{E}_x \left[\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)^\top\right]\hat{u}$. For notation simplicity, in this proof, we will temporarily denote $Z_i := \sigma\left(\frac{Vx_i+s}{\sqrt{2}}\right)$, $Z := \sigma\left(\frac{Vx+s}{\sqrt{2}}\right)$, $\Sigma := \mathbb{E}_x\left[ZZ^\top\right]$.

Noticing that we have

$$\frac{1}{n}\sum_{i=1}^n g_{\hat{u},s,V}(x_i)^2 \leqslant \frac{2}{n}\sum_{i=1}^n (g_{\hat{u},s,V}(x_i)-h(x_i))^2 + \frac{2}{n}\sum_{i=1}^n h(x_i)^2 \lesssim 1$$

with probability at least $1-\delta/64$, due to the small training loss and some standard concentration for $\frac{1}{n}\sum_i h(x_i)^2$. That is to say,

$$\hat{u}^\top\left(\frac{1}{n}\sum_{i=1}^n Z_iZ_i^\top\right)\hat{u} = \frac{1}{n}\sum_{i=1}^n\left(\hat{u}^\top Z_i\right)^2 = \frac{1}{n}\sum_{i=1}^n g_{\hat{u},s,V}(x_i)^2 \lesssim 1$$

Next, we bound the difference between $\hat{u}^\top\left(\frac{1}{n}\sum_{i=1}^n Z_iZ_i^\top\right)\hat{u}$ and $\hat{u}^\top\Sigma\hat{u}$. To this end, we orthogonally decompose Σ as $\Sigma = K^\top OK$, where O is a diagonal matrix and K is an orthogonal matrix. Write $O = \mathrm{diag}\{\gamma_1,\dots,\gamma_t,0,\dots,0\}$ for some integer $t = \mathrm{rank}(\Sigma)$, where $\gamma_i > 0$ for $i \in [t]$. Notice that $O^{1/2} = \mathrm{diag}\{\gamma_1^{1/2},\dots,\gamma_t^{1/2},0,\dots,0\}$, and we formally denote $O^{-1/2} = \mathrm{diag}\{\gamma_1^{-1/2},\dots,\gamma_t^{-1/2},0,\dots,0\}$. Due to the fact that $\mathbb{E}_x\left[KZZ^\top K^\top\right] = O$, we know KZ lies in the span of $\{e_1,\dots,e_t\}$. Therefore, we have

$$\begin{aligned}\left|\hat{u}^\top\left(\frac{1}{n}\sum_{i=1}^n Z_iZ_i^\top - \Sigma\right)\hat{u}\right| &= \left|\hat{u}^\top K^\top O^{1/2}\left(\frac{1}{n}\sum_{i=1}^n O^{-1/2}KZ_iZ_i^\top K^\top O^{-1/2} - \begin{pmatrix} I_t & \\ & 0\end{pmatrix}\right)O^{1/2}K\hat{u}\right| \\ &\leqslant \hat{u}^\top\Sigma\hat{u}\left\|\frac{1}{n}\sum_{i=1}^n O^{-1/2}KZ_iZ_i^\top K^\top O^{-1/2} - \begin{pmatrix} I_t & \\ & 0\end{pmatrix}\right\|\end{aligned}$$

Denote $W_i := O^{-1/2}KZ_i$ and $W := O^{-1/2}KZ$. We see that the second moment of $W_{\leqslant t}$ is equal to identity matrix in t dimensions: $\mathbb{E}_x\left[W_{\leqslant t}W_{\leqslant t}^\top\right] = I_t$. That is to say, $W_{\leqslant t}$ is isotropic. Next, we will bound the following operator norm

$$\left\|\frac{1}{n}\sum_{i=1}^n O^{-1/2}KZ_iZ_i^\top K^\top O^{-1/2} - \begin{pmatrix} I_t & \\ & 0\end{pmatrix}\right\| = \left\|\frac{1}{n}\sum_{i=1}^n W_{\leqslant t,i}W_{\leqslant t,i}^\top - I_t\right\|$$

by the following concentration lemma.

Lemma 18. *Let $W = W(x) \in \mathbb{R}^m$ be a random vector which is a function of $x \sim \gamma$. Assume for each $i \in [m]$, the i-th coordinate W_i is a k degree polynomial w.r.t. x. Also assume $\mathbb{E}_x\left[WW^\top\right] = I$. Let $W_1,\dots,W_n$ be i.i.d. generated samples. Then with probability at least $1-\delta/64$, we have*

$$\max_{1\leqslant j\leqslant m}\left|s_j(\widetilde{W}) - \sqrt{n}\right| \lesssim \sqrt{m\log m(\log n)^k}$$

where $\widetilde{W} = (W_1,\dots,W_n)^\top$ and s_j is the singular value.

Proof. For any $z \geqslant \sqrt{\mathrm{Var}(\|W\|^2)}$, we have the following estimation for the tail probability

$$\begin{aligned}\mathbb{P}\left(\max_{1\leqslant i\leqslant n}\|W_i\|^2 \geqslant z+m\right) &\leqslant n\mathbb{P}\left(\|W\|^2 \geqslant z+m\right) \\ &\leqslant n\mathbb{P}\left(\|W\|^2 - \mathbb{E}_x\left[\|W\|^2\right] \geqslant z\right) \\ &\leqslant 2n\exp\left(-\Theta(1)\left(\frac{z}{\sqrt{\mathrm{Var}(\|W\|^2)}}\right)^{1/k}\right)\end{aligned}$$

due to polynomial concentration, Corollory 3, where

$$\mathrm{Var}(\|W\|^2) \leqslant \mathbb{E}_x\left[\|W\|^4\right] \lesssim m\sum_{i=1}^{m}\mathbb{E}\left[W_i^4\right] \lesssim m\sum_{i=1}^{m}\left(\mathbb{E}\left[W_i^2\right]\right)^2 \lesssim m^2$$

Therefore, to estimate $\mathbb{E}\left[\max_{1\leqslant i\leqslant n}\|W_i\|^2\right]$, we can choose a truncation level $\Theta(1)(\log n)^k\sqrt{\mathrm{Var}(\|W\|^2)}+m$ with a large $\Theta(1)$.

$$\begin{aligned}
\mathbb{E}\left[\max_{1\leqslant i\leqslant n}\|W_i\|^2\right] &\lesssim m(\log n)^k + \mathbb{E}_x\left[\max_{1\leqslant i\leqslant n}\|W_i\|^2 1_{\max_{1\leqslant i\leqslant n}\|W_i\|^2\geqslant\Theta(1)(\log n)^k\sqrt{\mathrm{Var}(\|W\|^2)}+m}\right] \\
&\lesssim m(\log n)^k + \int_{\Theta(1)(\log n)^k\sqrt{\mathrm{Var}(\|W\|^2)}}^{+\infty} 2\exp\left(-\Theta(1)\left(\frac{z}{\sqrt{\mathrm{Var}(\|W\|^2)}}\right)^{1/k}+\log n\right)dz \\
&\lesssim m(\log n)^k + \int_{\Theta(1)\log n}^{+\infty}\exp(-\Theta(1)\tilde{z}+\log n)\tilde{z}^{k-1}d\tilde{z} \\
&\lesssim m(\log n)^k
\end{aligned}$$

We will use the above estimation and the following Lemma from Theorem 5.45, Vershynin (2010) to estimate the singular values of $\widetilde{W}$.

Lemma 19. *Let A be an $N\times n$ matrix whose rows A_i are independent isotropic random vectors in $\mathbb{R}^n$. Let $m := \mathbb{E}\max_{i\leqslant N}\|A_i\|_2^2$. Then*

$$\mathbb{E}\max_{j\leqslant n}\left|s_j(A)-\sqrt{N}\right| \lesssim \sqrt{m\log\min(N,n)}$$

Therefore, combining that lemma and Markov inequality to gain a high probability bound, with probability at least $1-\delta/64$, we have

$$\max_{1\leqslant j\leqslant m}\left|s_j(\widetilde{W})-\sqrt{n}\right| \lesssim \sqrt{m\log m(\log n)^k}$$

□

Applying Lemma 18 to $W_{\leqslant t}$, we have

$$\left\|\frac{1}{n}\sum_{i=1}^{n}W_{\leqslant t,i}W_{\leqslant t,i}^\top - I_t\right\| \lesssim \sqrt{\frac{t\log t(\log n)^k}{n}}$$

with probability at least $1-\delta/64$. Next, we give an upper bound over t, the rank of our kernel matrix Σ. Using the Hermite addition formula, we have

$$\sigma\left(\frac{Vx+s}{\sqrt{2}}\right) = \sum_{j=0}^{k}h_j(Vx)\odot A_j$$

where $A_j\in\mathbb{R}^m$ is some vector that only depends on $\sigma(\cdot)$, j and s. Plugging that in our Σ, we have the following decomposition

$$\begin{aligned}
\mathbb{E}_x\left[\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)\sigma\left(\frac{Vx+s}{\sqrt{2}}\right)^T\right] &= \mathbb{E}_x\left[\left(\sum_{j=0}^{k}h_j(Vx)\odot A_j\right)\left(\sum_{j=0}^{k}h_j(Vx)\odot A_j\right)^T\right] \\
&= \sum_{j=0}^{k}\mathbb{E}_x\left[(h_j(Vx)\odot A_j)(h_j(Vx)\odot A_j)^T\right] := \sum_{j=0}^{k}\Sigma_j
\end{aligned}$$

For each $0\leqslant j\leqslant k$, we have

$$\Sigma_j(p,q) = A_{j,p}A_{j,q}\langle v_p^{\otimes j}, v_q^{\otimes j}\rangle = \langle A_{j,p}v_p^{\otimes j}, A_{j,q}v_q^{\otimes j}\rangle$$

where $A_{j,l}$ is the l-th element of A_j, and $\Sigma_j(p,q)$ is the (p,q) element of our matrix Σ_j. Therefore, define $M_j = \left(A_{j,1}v_1^{\otimes j}, \ldots, A_{j,m}v_m^{\otimes j}\right) \in \mathbb{R}^{d^j \times m}$, and we have $\Sigma_j = M_j^T M_j$ and thus $\operatorname{rank}(\Sigma_j) \leqslant d^j$. Therefore, $\operatorname{rank}(\Sigma) \leqslant \sum_{j=0}^{k} \operatorname{rank}(\Sigma_j) \lesssim d^k$ and $t \lesssim d^k$.

Therefore, we have

$$\left\| \frac{1}{n} \sum_{i=1}^{n} W_{\leqslant t,i} W_{\leqslant t,i}^{\top} - I_t \right\| \lesssim \sqrt{\frac{t \log t (\log n)^k}{n}} \lesssim \sqrt{\frac{d^k \log d (\log n)^k}{n}}$$

and

$$\left| \hat{u}^{\top} \left(\frac{1}{n} \sum_{i=1}^{n} Z_i Z_i^{\top} - \Sigma \right) \hat{u} \right| \leqslant \hat{u}^{\top} \Sigma \hat{u} \left\| \frac{1}{n} \sum_{i=1}^{n} W_{\leqslant t,i} W_{\leqslant t,i}^{\top} - I_t \right\| \lesssim \sqrt{\frac{d^k \log d (\log n)^k}{n}} \hat{u}^{\top} \Sigma \hat{u}.$$

As a consequence, we have

$$\mathbb{E}\left[g_{\hat{u},s,V}(x)^2\right] = \hat{u}^{\top} \Sigma \hat{u} \lesssim \hat{u}^{\top} \left(\frac{1}{n} \sum_{i=1}^{n} Z_i Z_i^{\top} \right) \hat{u} \lesssim 1$$

when d is larger than some absolute constant. Recall that $\mathbb{E}_x\left[h(x)^2\right] = \mathcal{O}(1)$ and plug everything back into equation (17), we have

$$\left(\mathbb{E}_x \left[\left(\left(g_{\hat{u},s,V}(x) - h(x) \right)^2 \right) \mathbf{1}_{|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau} \right] \right)^2 \lesssim \mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau \right)$$

Therefore, we only need to bound the $\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau\right)$ by polynomial concentration. From Lemma 32, we get

$$\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \beta \sqrt{\operatorname{Var}(g_{\hat{u},s,V}(x) - h(x))} \right) \leqslant 2\exp(-\Theta(1)\beta^{2/r})$$

for any $\beta > 1$. Furthermore, notice that

$$\operatorname{Var}(g_{\hat{u},s,V}(x) - h(x)) \leqslant \mathbb{E}_x\left[(g_{\hat{u},s,V}(x) - h(x))^2\right] \lesssim \mathbb{E}\left[g_{\hat{u},s,V}(x)^2\right] + \mathbb{E}\left[h(x)^2\right] \lesssim 1$$

which is from the arguments above. Thus, for every $\tau \gtrsim 1$, we have

$$\mathbb{P}\left(|g_{\hat{u},s,V}(x) - h(x)| \geqslant \tau\right) \leqslant 2\exp\left(-\Theta(1)\tau^{2/r}\right)$$

and the proof is complete. □

D Proof of Theorem 1

At the end of the first stage, our learner is $h_{\theta^{(T_1)}} = g_{\hat{u},s,V}$. In the second stage of our training algorithm, letting $\hat{p} := g_{\hat{u},s,V}$, the network becomes

$$h_\theta(x) = \hat{p}(x) + \sum_{i=1}^{m_2} c_i \sigma_2(a_i \hat{p}(x) + b_i)$$

with a_i, b_i random and fixed and c_i trainable. The network thus implements 1-D kernel regression over the new input $\hat{p}$ in the second stage of our training algorithm.

By Corollary 2, with probability $1 - \delta/2$ we have

$$\|g_{\hat{u},s,V} - \mathcal{P}_k h\|_{L^2(\gamma)}^2 = \mathcal{O}((\log d)^{r/2} d^{-\alpha}) \text{ and } \left\|\mathcal{P}_k h - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right] p\right\|_{L^2(\gamma)}^2 = \widetilde{\mathcal{O}}(d^{-\alpha}).$$

For notational convenience, in the remainder of this section we let $\hat{p}$ be an arbitrary k degree polynomial satisfying the following assumption:

Assumption 7. *We have a k-degree polynomial $\hat{p}$ which satisfies*

$$\left\|\hat{p} - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right] p\right\|_{L^2(\gamma)}^2 = \mathcal{O}((\log d)^{r/2} d^{-\alpha})$$

where $\alpha \in (0,1)$. Also, recall that we have assumed $\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[g'(z)\right] = \Theta(1)$ and we denote this quantity as C_g.

To prove Theorem 1, we condition on the event that $\hat{p} = g_{\hat{u},V}$ satisfies this assumption, which occurs with probability $1 - \delta/2$.

In the following we may use $\sigma(\cdot)$ to denote $\sigma_2(\cdot)$, and use m to refer m_2, for notation simplicity. The proof strategy will be very similar with the proof in Appendix C. We begin by constructing a low-norm solution that obtains small loss. Next, we show GD converges to an approximate minimizer. We conclude by invoking Kernel Rademacher arguments to show generalization.

D.1 Approximation

Define $\widetilde{g}(z) = g(\frac{1}{C_g}z)$. The target can thus be represented as $\widetilde{g}(C_g p(x))$. We will proceed using the following two steps to bound the approximation error in $L^2(\gamma)$.

- Step I. Bound the difference between $\widetilde{g} \circ \hat{p}$ and $\widetilde{g} \circ (C_g p)$.
- Step II. Using a 1-D two-layer neural network to approximate the 1-D link function $\widetilde{g}$.

For step I, we have the following simple Lemma.

Lemma 20. *Under the assumptions above,* $\|\widetilde{g} \circ \hat{p} - h\|_{L^2(\gamma)}^2 = \mathcal{O}((\log d)^{r/2} d^{-\alpha})$.

Proof of Lemma 20. We have that

$$\begin{aligned}
\|\widetilde{g} \circ \hat{p} - \widetilde{g} \circ (C_g p)\|_{L^2(\gamma)}^2 &\lesssim \sum_{k=1}^{q} \left\|(\hat{p}(x))^k - (C_g p(x))^k\right\|_{L^2(\gamma)}^2 \\
&\leqslant \sum_{k=1}^{q} \mathbb{E}_x \left[(\hat{p}(x) - C_g p(x))^2 (\hat{p}(x)^{k-1} + \hat{p}(x)^{k-2}(C_g p(x)) + \cdots + (C_g p(x))^{k-1})^2\right] \\
&\leqslant \sum_{k=1}^{q} \sqrt{\mathbb{E}_x \left[(\hat{p}(x) - C_g p(x))^4\right] \mathbb{E}_x \left[(\hat{p}(x)^{k-1} + \hat{p}(x)^{k-2}(C_g p(x)) + \cdots + (C_g p(x))^{k-1})^4\right]} \\
&\lesssim \sum_{k=1}^{q} \mathbb{E}_x \left[(\hat{p}(x) - C_g p(x))^2\right] \mathbb{E}_x \left[(\hat{p}(x)^{k-1} + \hat{p}(x)^{k-2}(C_g p(x)) + \cdots + (C_g p(x))^{k-1})^2\right] \\
&\lesssim \|\hat{p} - C_g p\|_{L^2(\gamma)}^2 \lesssim (\log d)^{r/2} d^{-\alpha}
\end{aligned}$$

where the fourth inequality and the fifth inequality are due to Lemma 31, Gaussian hypercontractivity. We implicitly use $\|\hat{p}\|_{L^2(\gamma)} = \mathcal{O}(1)$ and $\|C_g p\|_{L^2(\gamma)} = \mathcal{O}(1)$ in the fifth inequality, too. □

Step II relies on Lemma 3, which is restated below:

Lemma 3. *Let* $m = d^\alpha$. *With probability at least* $1 - \delta/4$ *over the sampling of* a, b, *there exists some* c^* *such that* $\|c^*\|_\infty = \mathcal{O}((\log d)^{k(p+q)} d^{-\alpha})$ *and*

$$L(\theta^*) = \left\| \hat{p}(x) + \sum_{i=1}^{m} c_i^* \sigma(a_i \hat{p}(x) + b_i) - h(x) \right\|_{L^2(\gamma)}^2 = \mathcal{O}((\log d)^{r/2 + 2k(p+q)} d^{-\alpha})$$

Proof of Lemma 3. We will firstly control the typical value of $\hat{p}$. From Lemma 32, we have

$$\mathbb{P}\left[|\hat{p}(x)| \geqslant \beta \sqrt{\mathrm{Var}(\hat{p}(x))}\right] \leqslant 2 \exp\left(-\Theta(1) \min\left(\beta^2, \beta^{2/k}\right)\right)$$

for any $\beta > 0$. That is to say, when $\beta \geqslant 1$, with probability at least $1 - 2e^{-\Theta(1)\beta^{2/k}}$ we have $|\hat{p}(x)| \lesssim \beta$. We implicitly use $\|\hat{p}\|_{L^2(\gamma)} = \mathcal{O}(1)$ in this argument to bound $\mathrm{Var}(\hat{p}(x))$.

Next, we will use Lemma 39 to give a representation for $\widetilde{g}$ in the bounded domain. There exists $v(\cdot, \cdot)$ supported on $\{-1, 1\} \times [0, 2C\beta]$ such that for any x satisfying $|\hat{p}(x)| \leqslant C\beta$,

$$\mathbb{E}_{a,b}\left[v(a, b)\sigma(a\hat{p}(x) + b)\right] = \widetilde{g}(\hat{p}(x)) - \hat{p}(x)$$

where $a \sim \mathrm{Unif}\{-1,1\}$ and b has density $\mu_b(t)$. Furthermore, recall that we have assumed $\mu_b(t) \gtrsim (1+|t|)^{-p}$, and we have the following estimation $\sup_{a,b}|v(a,b)| = \mathcal{O}(\beta^{p+q})$.

Next, we will do a Monte Carlo sampling to approximate the target.

$$\begin{aligned}
&\mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \\
&\leqslant \mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \mathbf{1}_{|\hat{p}(x)|\geqslant C\beta} \\
&\quad + \mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \mathbf{1}_{|\hat{p}(x)|\leqslant C\beta}
\end{aligned} \tag{19}$$

For the second term, we have

$$\begin{aligned}
&\mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \mathbf{1}_{|\hat{p}(x)|\leqslant C\beta} \\
&\leqslant \mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-\mathbb{E}_{a,b}\left[v(a,b)\sigma(a\hat{p}(x)+b)\right]\right)^2 \\
&\leqslant \frac{1}{m}\,\mathbb{E}_x\,\mathbb{E}_{a,b}\left(v(a,b)\sigma(a\hat{p}(x)+b)\right)^2 \\
&\leqslant \frac{1}{m}\mathcal{O}(\beta^{2p+2q})\left(\mathbb{E}_x\,\hat{p}(x)^2+\mathbb{E}_b\,b^2\right) = \frac{1}{m}\mathcal{O}(\beta^{2p+2q})
\end{aligned} \tag{20}$$

Here we implicitly use the fact that $\mathbb{E}_b\,b^2 = \mathcal{O}(1)$ which is from our assumptions on $\mu_b(t)$. For the first term, by Cauchy inequality,

$$\begin{aligned}
&\mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \mathbf{1}_{|\hat{p}(x)|\geqslant C\beta} \\
&\leqslant \sqrt{\mathbb{E}_{a,b,x}\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^4 \mathbb{P}\left(|\hat{p}(x)|\geqslant C\beta\right)} \\
&\lesssim e^{-\Theta(1)\beta^{2/k}}\sqrt{\mathbb{E}_{a,b,x}\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^4} \\
&\lesssim e^{-\Theta(1)\beta^{2/k}}\sqrt{\mathbb{E}_{a,b,x}\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)\right)^4+\mathbb{E}_x(\widetilde{g}(\hat{p}(x))-\hat{p}(x))^4} \\
&\lesssim e^{-\Theta(1)\beta^{2/k}}\sqrt{\mathbb{E}_{a,b,x}\left(v(a,b)\sigma(a\hat{p}(x)+b)\right)^4+\mathcal{O}(1)} \\
&\lesssim e^{-\Theta(1)\beta^{2/k}}\beta^{2p+2q}
\end{aligned}$$

Here we implicitly use the fact that $\mathbb{E}_b\,b^4 = \mathcal{O}(1)$ which is again from our assumptions on $\mu_b(t)$. We also use gaussian hypercontractivity, Lemma 31 to show $\mathbb{E}_x(\widetilde{g}(\hat{p}(x))-\hat{p}(x))^4 = \mathcal{O}(1)$. Since $\hat{p}(x)$ is a k degree polynomial with Gaussian input distribution, its higher order moments can be bounded by a polynomial of its second moment which is clearly $\mathcal{O}(1)$.

From the above arguments, we already derive

$$\mathbb{E}_{a,b}\,\mathbb{E}_x\left(\frac{1}{m}\sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i)-(\widetilde{g}(\hat{p}(x))-\hat{p}(x))\right)^2 \lesssim \left(\frac{1}{m}+e^{-\Theta(1)\beta^{2/k}}\right)\beta^{2p+2q}$$

Therefore, for any absolute constant $\delta \in (0,1)$, with probability at least $1-\delta/4$ over the sampling of the random features a_i, b_i, using Markov inequality, we have

$$\mathbb{E}_x \left(\frac{1}{m} \sum_{i=1}^m v(a_i,b_i)\sigma(a_i\hat{p}(x)+b_i) - (\widetilde{g}(\hat{p}(x)) - \hat{p}(x)) \right)^2 \lesssim \left(\frac{1}{m} + e^{-\Theta(1)\beta^{2/k}} \right) \beta^{2p+2q}$$

Combining this with our previous result, Lemma 20, with probability at least $1-\delta/4$ over the sampling of the random features, we can find the parameters c^* in the third layer with $\sup_i |c_i^*| = \mathcal{O}(\beta^{p+q}/m)$, such that

$$L(\theta^*) = \left\| \hat{p}(x) + \sum_{i=1}^m c_i^* \sigma(a_i\hat{p}(x)+b_i) - h(x) \right\|_{L^2(\gamma)}^2 \lesssim \left(\frac{1}{m} + e^{-\Theta(1)\beta^{2/k}} \right) \beta^{2p+2q} + (\log d)^{r/2} d^{-\alpha}$$

where $\theta^* = (a^{(0)}, b^{(0)}, c^*, \hat{u}, V^{(0)})$. Let us further set $\beta = \Theta(1)(\log d)^k$ where $\Theta(1)$ is some large absolute constant. Set $m = d^\alpha$. In this case, we will have

$$L(\theta^*) \lesssim (d^{-\alpha} + e^{-\log^2 d})(\log d)^{2k(p+q)} + (\log d)^{r/2} d^{-\alpha} \lesssim (\log d)^{r/2+2k(p+q)} d^{-\alpha}$$

□

D.2 Empirical Performance

Next we will show the existence of good estimators in our empirical landscape. Firstly, we need to concentrate the landscape at the special point c^* we constructed. With a little abuse of notations, denote the empirical version of the square loss as

$$\hat{L}(\theta) = \frac{1}{n} \sum_{j=1}^n \left(\hat{p}(x_j) + \sum_{i=1}^m c_i \sigma(a_i \hat{p}(x_j) + b_i) - h(x_j) \right)^2$$

where we recall that $x_j \in \mathcal{D}_2$ is newly generated data which is independent of $\mathcal{D}_1$.

Lemma 21. *With probability at least $1 - 3\delta/8 - \mathcal{O}(1)d^{-\alpha}$, we will have*

$$\hat{L}(\theta^*) \leqslant \frac{1}{\sqrt{n}} \mathcal{O}((\log d)^{2k(p+q)}) + \mathcal{O}((\log d)^{r/2+2k(p+q)} d^{-\alpha})$$

Proof of Lemma 21. In the following, we compute the variance term.

$$\begin{aligned}
\mathbb{E}_x \left(\hat{L}(\theta^*) - L(\theta^*) \right)^2 &= \frac{1}{n} \operatorname{Var} \left(\left(\sum_{i=1}^m c_i^* \sigma(a_i\hat{p}(x)+b_i) - (h(x) - \hat{p}(x)) \right)^2 \right) \\
&\leqslant \frac{1}{n} \mathbb{E}_x \left(\sum_{i=1}^m c_i^* \sigma(a_i\hat{p}(x)+b_i) - (h(x) - \hat{p}(x)) \right)^4 \\
&\lesssim \frac{1}{n} \left(\mathbb{E}_x \left(\sum_{i=1}^m c_i^* \sigma(a_i\hat{p}(x)+b_i) \right)^4 + \mathbb{E}_x (h(x))^4 + \mathbb{E}_x\, \hat{p}(x)^4 \right) \\
&\leqslant \frac{1}{n} \left(m^3 \sum_{i=1}^m \mathbb{E}_x\, c_i^{*4} (a_i\hat{p}(x)+b_i)^4 + \mathcal{O}(1) \right) \\
&\lesssim \frac{1}{n} \left(1 + \beta^{4p+4q} \frac{1}{m} \sum_{i=1}^m (b_i^4 + \mathbb{E}_x\, \hat{p}(x)^4) \right) \\
&\lesssim \frac{1}{n} \beta^{4p+4q} \left(1 + \frac{1}{m} \sum_{i=1}^m b_i^4 \right)
\end{aligned}$$

Here are some technical arguments to bound $\frac{1}{m}\sum_{i=1}^m b_i^4$. We have

$$\mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^m b_i^4 - \mathbb{E}_b\, b^4\right)^2 \leqslant \frac{1}{m}\,\mathbb{E}_b\, b^8$$

and

$$\mathbb{P}_b\left(\left(\frac{1}{m}\sum_{i=1}^m b_i^4 - \mathbb{E}_b\, b^4\right)^2 \geqslant 1\right) \leqslant \mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^m b_i^4 - \mathbb{E}_b\, b^4\right)^2 \leqslant \frac{1}{m}\,\mathbb{E}_b\, b^8$$

Therefore, recall that $\mathbb{E}_b\, b^8 = \mathcal{O}(1)$ based on our assumption on $\mu_b(t)$, we will have with probability $1 - \mathcal{O}(1)d^{-\alpha}$, $\frac{1}{m}\sum_{i=1}^m b_i^4 \lesssim 1$. In that case, we have

$$\mathbb{E}_x\left(\hat{L}(\theta^*) - L(\theta^*)\right)^2 \lesssim \frac{1}{n}\beta^{4p+4q} = \frac{1}{n}(\log d)^{4k(p+q)}$$

Therefore, by Markov inequality, we have $\left|\hat{L}(\theta^*) - L(\theta^*)\right| \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)}$ with probability at least $1 - \delta/8$. In this case, we have

$$\hat{L}(\theta^*) \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + (\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

□

In the second stage of our training algorithm, we are doing the following minimization problem

$$\min_c \hat{L}(\theta) + \frac{1}{2}\xi_2\,\|c\|^2$$

via vanilla GD, where $\theta = (a^{(0)}, b^{(0)}, c, \hat{u}, V^{(0)})$. Since this problem is strongly convex and smooth, the optimization problem can be easily solved by plain GD.

Lemma 22. *Set $\xi_2 = 2$. For any $\epsilon \in (0,1)$, let $T_2 \gtrsim m\log(m/\epsilon)$. Then, when m, n, d are larger than some absolute constant, with probability at least $1 - 7\delta/16$, the predictor $\hat{c} := c^{(T_2)}$ and $\hat{\theta} = (a^{(0)}, b^{(0)}, \hat{c}, \hat{u}, V^{(0)})$ satisfies*

$$\hat{L}(\hat{\theta}) \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + \epsilon + (\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

and

$$\|\hat{c}\|^2 \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + \epsilon + (\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

Proof. For any given threshold $\epsilon \in (0,1)$, assuming $\hat{c}$ is an ϵ minimizer of the optimization problem, then we will have

$$\hat{L}(\hat{\theta}) + \frac{1}{2}\xi_2\,\|\hat{c}\|^2 \leqslant \hat{L}(\theta^*) + \frac{1}{2}\xi_2\,\|c^*\|^2 + \epsilon \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + \epsilon + (1+\xi_2)\,(\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

Plug $\xi_2 = 2$ in, then we will have

$$\hat{L}(\hat{\theta}) \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + \epsilon + (\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

and

$$\|\hat{c}\|^2 \lesssim \frac{1}{\sqrt{n}}(\log d)^{2k(p+q)} + \epsilon + (\log d)^{r/2+2k(p+q)}d^{-\alpha}$$

It thus suffices to analyze the optimization problem.

Clearly, this convex optimization problem is at least 2-strongly convex. To estimate the time complexity, we also need to estimate the smoothness of our optimization objective.

Lemma 23. *With probability at least $1 - \mathcal{O}(1)d^{-\alpha} - 2e^{-\Theta(1)n^{1/2k}}$, we have*

$$\left|\nabla\hat{L}(c_1) - \nabla\hat{L}(c_2)\right| \lesssim m$$

Proof. We first calculate the gradient

$$\nabla \hat{L}(\theta) = \frac{2}{n}\sum_{j=1}^{n}\left(\hat{p}(x_j) + c^\top \sigma(a\hat{p}(x_j)+b) - h(x_j)\right)\sigma(a\hat{p}(x_j)+b)$$

then bound the Lipschitz constant for the gradient

$$\begin{aligned}\left|\nabla \hat{L}(c_1) - \nabla \hat{L}(c_2)\right| &= \left|\frac{2}{n}\sum_{j=1}^{n}\langle c_1 - c_2, \sigma(a\hat{p}(x_j)+b)\rangle \sigma(a\hat{p}(x_j)+b)\right| \\ &\leqslant \frac{2}{n}\sum_{j=1}^{n}\|c_1-c_2\|\,\|\sigma(a\hat{p}(x_j)+b)\|^2 \\ &\leqslant \|c_1-c_2\|\left(\frac{2}{n}\sum_{j=1}^{n}\sum_{i=1}^{m}(a_i\hat{p}(x_j)+b_i)^2\right) \\ &\leqslant \|c_1-c_2\|\left(\frac{4m}{n}\sum_{j=1}^{n}\hat{p}(x_j)^2 + 4\sum_{i=1}^{m}b_i^2\right)\end{aligned}$$

Here are some technical arguments to estimate $\sum_i b_i^2$. We have

$$\mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^{m}b_i^2 - \mathbb{E}_b\, b^2\right)^2 \leqslant \frac{1}{m}\,\mathbb{E}_b\, b^4$$

and

$$\mathbb{P}_b\left(\left(\frac{1}{m}\sum_{i=1}^{m}b_i^2 - \mathbb{E}_b\, b^2\right)^2 \geqslant 1\right) \leqslant \mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^{m}b_i^2 - \mathbb{E}_b\, b^2\right)^2 \leqslant \frac{1}{m}\,\mathbb{E}_b\, b^4$$

Therefore, recall that $m = d^\alpha$, and also $\mathbb{E}_b\, b^4 = \mathcal{O}(1)$ due to our assumption on $\mu_b(t)$, we will have with probability $1 - \mathcal{O}(1)d^{-\alpha}$, $\frac{1}{m}\sum_{i=1}^{m}b_i^2 \lesssim 1$. Moreover, we can use Corollary 3 to concentrate $\sum_j \hat{p}(x_j)^2$. More concretely, we will have $\frac{1}{n}\sum_j \hat{p}(x_j)^2 \lesssim 1$ with probability at least $1 - 2e^{-\Theta(1)n^{1/2k}}$, since $\hat{p}(x)^2$ is a degree $2k$ polynomial and $\mathrm{Var}(\hat{p}(x)^2) \lesssim 1$ via Gaussian hypercontractivity, Lemma 31. Therefore, with probability at least $1 - \mathcal{O}(1)d^{-\alpha} - 2e^{-\Theta(1)n^{1/2k}}$, we have

$$\left|\nabla \hat{L}(c_1) - \nabla \hat{L}(c_2)\right| \lesssim 1$$

□

Having derived the above Lemma, using Lemma 36 in Appendix E.5, we can choose the learning rate $\eta_1 = \frac{1}{\Theta(m)}$ and have

$$\left\|c^{(t)} - c_{opt}\right\|^2 \leqslant \left(1 - \frac{1}{\Theta(m)}\right)^t \|c_{opt}\|^2$$

where c_{opt} is the unique optimal solution for that optimization problem. Furthermore, we have the following

$$\begin{aligned}\sup_{\|c\|\leqslant R}\left\|\nabla \hat{L}(c) + 2c\right\| &\leqslant \sup_{\|c\|\leqslant R}\left\|\nabla \hat{L}(c)\right\| + 2R \\ &\leqslant \frac{2}{n}\sum_{j=1}^{n}\|\sigma(a\hat{p}(x_j)+b)\|\left(|\hat{p}(x_j) - h(x_j)| + R\,\|\sigma(a\hat{p}(x_j)+b)\|\right) + 2R \\ &\leqslant \frac{2}{n}\sum_{j=1}^{n}\left((R+1)\,\|\sigma(a\hat{p}(x_j)+b)\|^2 + (\hat{p}(x_j) - h(x_j))^2\right) + 2R \\ &\leqslant (R+1)\mathcal{O}(m) + \frac{2}{n}\sum_{j=1}^{n}(\hat{p}(x_j) - h(x_j))^2 + 2R\end{aligned}$$

with probability at least $1 - \mathcal{O}(1)d^{-\alpha} - 2e^{-\Theta(1)n^{1/2k}}$. The last inequality follows from the same argument in Lemma 23. Moreover, we can use Corollary 3 to concentrate $\sum_j (\hat{p}(x_j) - h(x_j))^2$. More concretely, we will have $\frac{1}{n}\sum_j (\hat{p}(x_j) - h(x_j))^2 \lesssim 1$ with probability at least $1 - 2e^{-\Theta(1)n^{1/2r}}$, since $(\hat{p}(x) - h(x))^2$ is a degree $2r$ polynomial and $\mathrm{Var}\left((\hat{p}(x) - h(x))^2\right) \lesssim 1$ via Gaussian hypercontractivity, Lemma 31. Therefore, with probability at least $1 - \mathcal{O}(1)d^{-\alpha} - 2e^{-\Theta(1)n^{1/2r}}$, we have

$$\sup_{\|c\| \leqslant R} \left\| \nabla \hat{L}(c) + 2c \right\| \lesssim (R+1)m$$

Utilizing that fact, we have

$$\hat{L}(c^{(t)}) + \left\| c^{(t)} \right\|^2 \leqslant \hat{L}(c_{opt}) + \|c_{opt}\|^2 + \sup_{\|c\| \leqslant 2\|c_{opt}\|} \left\| \nabla \hat{L}(c) + 2c \right\| \left\| c^{(t)} - c_{opt} \right\|$$

Since $\|c_{opt}\| = \mathcal{O}(1)$, $\sup_{\|c\| \leqslant 2\|c_{opt}\|} \left\| \nabla \hat{L}(c) + 2c \right\| = \mathcal{O}(m)$, if we want

$$\sup_{\|c\| \leqslant 2\|c_{opt}\|} \left\| \nabla \hat{L}(c) + 2c \right\| \left\| c^{(t)} - c_{opt} \right\| \leqslant \epsilon_2$$

it is sufficient to have $T_2 \gtrsim m \log(m/\epsilon_2)$. □

In addition, for any truncation level $\tau > 0$, we will also have

$$\frac{1}{n}\sum_{j=1}^{n} \ell_\tau(h_{\hat{\theta}}(x_j), h(x_j)) \leqslant \hat{L}(\hat{\theta}) \lesssim \frac{1}{\sqrt{n}} (\log d)^{2k(p+q)} + \epsilon + (\log d)^{r/2 + 2k(p+q)} d^{-\alpha}$$

which we will use later. Here we recall $\ell_\tau(x, y) := (x-y)^2 \wedge \tau^2$.

D.3 Uniform Generalization Bounds

To conclude, we need a uniform generalization bound over c for our population loss $L(\theta) = \|h_\theta - h\|^2_{L^2(\gamma)}$. As in Appendix C, we bound the truncated loss via a Rademacher complexity argument, and deal with the truncation term later.

Proof of Theorem 1. Recall that $\ell_\tau(x, y) = (x-y)^2 \wedge \tau^2$. From Lemma 33 and 34, with probability at least $1 - \delta/32$, we will have

$$\sup_{\|c\| \leqslant M_c} \left| \frac{1}{n}\sum_{i=1}^{n} \ell_\tau(h_\theta(x_i), h(x_i)) - \mathbb{E}_x \left[\ell_\tau(h_\theta(x), h(x)) \right] \right| \leqslant 4\tau \,\mathrm{Rad}_n(\mathcal{H}) + \tau^2 \sqrt{\frac{\mathcal{O}(1)}{n}}$$

where $\mathcal{H} := \{h_\theta : \|c\| \leqslant M_c\}$. Then we will compute $\mathrm{Rad}_n(\mathcal{H})$.

Lemma 24. *With probability at least $1 - \mathcal{O}(1)d^{-\alpha}$ over the sampling of a, b, we have*

$$\mathrm{Rad}_n(\mathcal{H}) \lesssim M_c \sqrt{\frac{m}{n}}$$

Proof.

$$\begin{aligned}
\mathrm{Rad}_n(\mathcal{H}) &= \mathbb{E}_x\,\mathbb{E}_\xi\left[\sup_{\|c\|\leqslant M_c}\frac{1}{n}\sum_{j=1}^n\xi_j\left(\sum_{i=1}^m c_i\sigma(a_i\hat{p}(x_j)+b_i)\right)\right]\\
&= \frac{1}{n}\,\mathbb{E}_x\,\mathbb{E}_\xi\left[\sup_{\|c\|\leqslant M_c}\sum_{i=1}^m c_i\left(\sum_{j=1}^n\xi_j\sigma(a_i\hat{p}(x_j)+b_i)\right)\right]\\
&\leqslant \frac{M_c}{n}\,\mathbb{E}_x\,\mathbb{E}_\xi\sqrt{\sum_{i=1}^m\left(\sum_{j=1}^n\xi_j\sigma(a_i\hat{p}(x_j)+b_i)\right)^2}\\
&\leqslant \frac{M_c}{n}\sqrt{\mathbb{E}_x\,\mathbb{E}_\xi\sum_{i=1}^m\left(\sum_{j=1}^n\xi_j\sigma(a_i\hat{p}(x_j)+b_i)\right)^2}\\
&= \frac{M_c}{n}\sqrt{\mathbb{E}_x\left[\sum_{i=1}^m\sum_{j=1}^n\left(\sigma(a_i\hat{p}(x_j)+b_i)\right)^2\right]}\\
&\lesssim \frac{M_c}{\sqrt{n}}\sqrt{m\,\mathbb{E}_x\,\hat{p}(x)^2+\sum_{i=1}^m b_i^2}
\end{aligned}$$

Here are some technical arguments to estimate $\sum_i b_i^2$. We have

$$\mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^m b_i^2-\mathbb{E}_b\,b^2\right)^2\leqslant\frac{1}{m}\,\mathbb{E}_b\,b^4$$

and

$$\mathbb{P}_b\left(\left(\frac{1}{m}\sum_{i=1}^m b_i^2-\mathbb{E}_b\,b^2\right)^2\geqslant 1\right)\leqslant\mathbb{E}_b\left(\frac{1}{m}\sum_{i=1}^m b_i^2-\mathbb{E}_b\,b^2\right)^2\leqslant\frac{1}{m}\,\mathbb{E}_b\,b^4$$

Therefore, recall that $m=d^\alpha$, and also $\mathbb{E}_b\,b^4\lesssim 1$ due to our assumption on $\mu_b(t)$, we will have with probability $1-\mathcal{O}(1)d^{-\alpha}$, $\frac{1}{m}\sum_{i=1}^m b_i^2\lesssim 1$. In that case, plugging that in, we get our Lemma. □

As a consequence, with probability at least $1-\delta/32-\mathcal{O}(1)d^{-\alpha}$,

$$\sup_{\|c\|\leqslant M_c}\left|\frac{1}{n}\sum_{i=1}^n\ell_\tau(h_\theta(x_i),h(x_i))-\mathbb{E}_x\left[\ell_\tau(h_\theta(x),h(x))\right]\right|\lesssim 4\tau M_c\sqrt{\frac{m}{n}}+\tau^2\sqrt{\frac{1}{n}}$$

Lastly, we also need to deal with the truncation to get a L^2 generalization bound. That is to say, we need to bound

$$\sup_{\|c\|\leqslant M_c}\mathbb{E}_x\left[(h_\theta(x)-h(x))^2\mathbf{1}_{|h_\theta(x)-h(x)|\geqslant\tau}\right]$$

Lemma 25. *We will have with probability at least* $1-\mathcal{O}(1)d^{-\alpha}$*,*

$$\sup_{\|c\|\leqslant M_c}\mathbb{E}_x\left[(h_\theta(x)-h(x))^2\mathbf{1}_{|h_\theta(x)-h(x)|\geqslant\tau}\right]\lesssim\frac{1}{\tau^2}(1+m^4M_c^4)$$

Proof. By Cauchy inequality, we have

$$\begin{aligned}
\left(\mathbb{E}_x\left[\left((h_\theta(x)-h(x))^2\right)\mathbf{1}_{|h_\theta(x)-h(x)|\geqslant\tau}\right]\right)^2&\leqslant\mathbb{E}_x\left[(h_\theta(x)-h(x))^4\right]\mathbb{P}\left(|h_\theta(x)-h(x)|\geqslant\tau\right)\\
&\lesssim\left(\mathbb{E}_x\left[h_\theta(x)^4\right]+\mathbb{E}_x\left[h(x)^4\right]\right)\mathbb{P}\left(|h_\theta(x)-h(x)|\geqslant\tau\right)
\end{aligned}\tag{21}$$

Recall that $\mathbb{E}_x h(x)^4 = \mathcal{O}(1)$. In addition, we have

$$\begin{aligned}\mathbb{E}_x\left[h_\theta(x)^4\right] &= \mathbb{E}_x\left[\left(\sum_{i=1}^m c_i\sigma(a_i\hat{p}(x)+b_i)\right)^4\right]\\ &\leqslant m^3\sum_{i=1}^m \mathbb{E}_x\left[c_i^4(a_i\hat{p}(x)+b_i)^4\right]\\ &\lesssim m^4M_c^4\left(\mathcal{O}(1)+\frac{1}{m}\sum_{i=1}^m b_i^4\right)\lesssim m^4M_c^4\end{aligned}$$

if under the high probability event $\frac{1}{m}\sum_{i=1}^m b_i^4 \lesssim 1$. Furthermore, we have

$$\mathbb{P}\left(|h_\theta(x)-h(x)|\geqslant\tau\right)\leqslant\frac{1}{\tau^4}\mathbb{E}_x\left[(h_\theta(x)-h(x))^4\right]\lesssim\frac{1}{\tau^4}(1+m^4M_c^4)$$

Plugging this back, we will have with probability at least $1-\mathcal{O}(1)d^{-\alpha}$,

$$\sup_{\|c\|\leqslant M_c}\mathbb{E}_x\left[(h_\theta(x)-h(x))^2\mathbf{1}_{|h_\theta(x)-h(x)|\geqslant\tau}\right]\lesssim\frac{1}{\tau^2}(1+m^4M_c^4)$$

□

We now combine everything together. Let us choose $\epsilon = d^{-\alpha}$ and $n \geqslant d^{k+3\alpha}$ and recall $m = d^\alpha$. In that case, $\|\hat{c}\|^2 = \mathcal{O}((\log d)^{r/2+2k(p+q)}d^{-\alpha})$. Therefore, when d is larger than some constant that is only depending on r, p, α, we are allowed to set $M_c = (\log d)^{\Theta(1)}d^{-\alpha}$ for some large $\Theta(1)$. In that case, we have

$$\left\|h_{\hat{\theta}}-h\right\|_{L^2(\gamma)}^2 \lesssim (\log d)^{r/2+2k(p+q)}d^{-\alpha}+4\tau(\log d)^{\Theta(1)}d^{-\alpha}\sqrt{d^{-k-2\alpha}}+\tau^2d^{-k/2-3\alpha/2}+\tau^{-2}(\log d)^{\Theta(1)}$$

We will pick up our truncation level $\tau = d^{\alpha/2}$. In that case, for any $\alpha \in (0,1)$, we will have

$$\left\|h_{\hat{\theta}}-h\right\|_{L^2(\gamma)}^2 = \mathcal{O}((\log d)^{\Theta(1)}d^{-\alpha}) = \widetilde{\mathcal{O}}(d^{-\alpha})$$

□

E Technical Background

E.1 Hermite Polynomials

Definition 3 (1D Hermite polynomials)**.** The k-th normalized probabilist's Hermite polynomial, $h_k : \mathbb{R} \to \mathbb{R}$, is the degree k polynomial defined as

$$h_k(x) = \frac{(-1)^k}{\sqrt{k!}}\frac{\frac{d^k\mu_\beta}{dx^k}(x)}{\mu_\beta(x)}, \tag{22}$$

where $\mu_\beta(x) = \exp(-x^2/2)/\sqrt{2\pi}$ is the density of the standard Gaussian.

The first such Hermite polynomials are

$$h_0(z) = 1, h_1(z) = z, h_2(z) = \frac{z^2-1}{\sqrt{2}}, h_3(z) = \frac{z^3-3z}{\sqrt{6}}, \cdots$$

Denote $\beta = \mathcal{N}(0,1)$ to be the standard Gaussian in 1D. A key fact is that the normalized Hermite polynomials form an orthonormal basis of $L^2(\beta)$; that is $\mathbb{E}_{x\sim\beta}[h_j(x)h_k(x)] = \delta_{jk}$.

Given a $f \in L^2(\beta)$, denote by $f(z) = \sum_k \hat{f}_k h_k(z)$ be the Hermite expansion of f where

$$\hat{f}_k = \mathbb{E}_{z\sim\beta}\left[f(z)h_k(z)\right] = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} f(z)h_k(z)e^{-\frac{z^2}{2}}\mathrm{d}z$$

is the Hermite coefficient of f. The following lemma will be useful, which can be found in Proposition 11.31 of O'Donnell (2014).

Lemma 26. *Given $f, g \in L^2(\beta)$, we have for any $u, v \in \mathbb{S}^{d-1}$ that*

$$\mathbb{E}_{x\sim\gamma}\left[f(u^\top x)g(v^\top x)\right] = \sum_{k=0}^{\infty} \hat{f}_k \hat{g}_k (u^\top v)^k$$

The multidimensional analog of the Hermite polynomials is *Hermite tensors*:

Definition 4 (Hermite tensors)**.** The k-th Hermite tensor in dimension d, $He_k : \mathbb{R}^d \to (\mathbb{R}^d)^{\otimes k}$, is defined as

$$He_k(x) = \frac{(-1)^k}{\sqrt{k!}} \frac{\nabla^k \mu_\gamma(x)}{\mu_\gamma(x)},$$

where $\mu_\gamma(x) = \exp(-\frac{1}{2}\|x\|^2)/(2\pi)^{d/2}$ is the density of the d-dimensional standard Gaussian.

The Hermite tensors form an orthonormal basis of $L^2(\gamma)$; that is, for any $f \in L^2(\gamma)$, one can write the Hermite expansion

$$f(x) = \sum_{k\geqslant 0} \langle C_k(f), He_k(x)\rangle \quad \text{where} \quad C_k(f) := \mathbb{E}_{x\sim\gamma}[f(x)He_k(x)].$$

We define the Hermite projection operator as $(\mathcal{P}_k f)(x) := \langle C_k(f), He_k(x)\rangle$. Intuitively speaking, the operator $\mathcal{P}_k$ extracts out the k degree part of a function when the input distribution is standard Gaussian. Furthermore, denote $\mathcal{P}_{\leqslant k} := \sum_{0\leqslant i\leqslant k} \mathcal{P}_i$ and $\mathcal{P}_{<k} := \sum_{0\leqslant i<k} \mathcal{P}_i$ as the projection operator onto the span of Hermite polynomials with degree no more than k, and degree less than k. It is clear that $\|\mathcal{P}_{\leqslant k} f\|_{L^2} \leqslant \|f\|_{L^2}$ for any $f \in L^2(\gamma)$. This can be shown by a simple Hermite expansion for f.

The next lemma can be shown by direct verification.

Lemma 27. *We have*

$$He_k(x) = \frac{1}{\sqrt{k!}} \mathbb{E}_{z\sim\gamma}\left[(x+iz)^{\otimes k}\right].$$

Lemma 28. *If $\|u\| = 1$, we have*

$$h_k(u^\top x) = \left\langle He_k(x), u^{\otimes k}\right\rangle.$$

Proof.

$$\begin{aligned}
\left\langle He_k(x), u^{\otimes k}\right\rangle &= \frac{1}{\sqrt{k!}} \left\langle \mathbb{E}_{z\sim\gamma}\left[(x+iz)^{\otimes k}\right], u^{\otimes k}\right\rangle \\
&= \frac{1}{\sqrt{k!}} \mathbb{E}_{z\sim\gamma}\left[(u^\top x + i(u^\top z))^k\right] \\
&= \frac{1}{\sqrt{k!}} \mathbb{E}_{z\sim\beta}\left[(u^\top x + iz)^k\right] = h_k(u^\top x).
\end{aligned}$$

□

E.2 Gaussian Hypercontractivity

By Holder's inequality, we have $\|X\|_{L^p} \leqslant \|X\|_{L^q}$ for any random variable X and any $p \leqslant q$. The reverse inequality does not hold in general, even up to a constant. However, for some measures like Gaussian, the reverse inequality will hold for some sufficiently nice functions like polynomials. The following lemma comes from Lemma 20 in Mei et al. (2021).

Lemma 29. *For any $\ell \in \mathbb{N}$ and $f \in L^2(\beta)$ to be a degree ℓ polynomial on $\mathbb{R}$ where β is the standard Gaussian distribution, for any $q \geqslant 2$, we have*

$$\left(\mathbb{E}_{z\sim\beta}\left[f(z)^q\right]\right)^{2/q} \leqslant (q-1)^\ell \mathbb{E}_{z\sim\beta}\left[f(z)^2\right]$$

The next Lemma is also from Mei et al. (2021) and is designed for uniform distribution on the sphere in d dimension.

Lemma 30. *For any $\ell \in \mathbb{N}$ and $f \in L^2(\mathbb{S}^{d-1})$ to be a degree ℓ polynomial, for any $q \geqslant 2$, we have*

$$\left(\mathbb{E}_{z\sim \mathrm{Unif}(\mathbb{S}^{d-1})}\left[f(z)^q\right]\right)^{2/q} \leqslant (q-1)^\ell \, \mathbb{E}_{z\sim \mathrm{Unif}(\mathbb{S}^{d-1})}\left[f(z)^2\right]$$

For the case where the input distribution is standard Gaussian in d dimension, we shall use the next Lemma from Theorem 4.3, Prato & Tubaro (2007).

Lemma 31. *For any $\ell \in \mathbb{N}$ and $f \in L^2(\gamma)$ to be a degree ℓ polynomial, for any $q \geqslant 2$, we have*

$$\mathbb{E}_{z\sim\gamma}\left[f(z)^q\right] \leqslant \mathcal{O}_{q,\ell}(1)\left(\mathbb{E}_{z\sim\gamma}\left[f(z)^2\right]\right)^{q/2}$$

where we use $\mathcal{O}_{q,\ell}(1)$ to denote some universal constant that only depends on q, ℓ.

E.3 Polynomial Concentration

In this subsection, we will introduce several Lemmas to control the deviation of random variables which polynomially depend on some Gaussian random variables. We will use a slightly modified version of Lemma 30 from Damian et al. (2022).

Lemma 32. *Let g be a polynomial of degree p and $x \sim \mathcal{N}(0, I_d)$. Then there exists an absolute positive constant C_p depending only on p such that for any $\delta > 0$,*

$$\mathbb{P}\left[|g(x) - \mathbb{E}[g(x)]| \geqslant \delta\sqrt{\mathrm{Var}(g(x))}\right] \leqslant 2\exp\left(-C_p \min\left(\delta^2, \delta^{2/p}\right)\right)$$

Consider the case that $x = (x_1, \ldots, x_n)$ and $g(x) = \frac{1}{n}\sum_i g(x_i)$, $x_i \sim_{i.i.d.} \mathcal{N}(0, I_d) \in \mathbb{R}^d$ and $x \in \mathbb{R}^{d\times n}$. Plug them into the above Lemma, and we get the following corollary.

Corollary 3. *Let g be a polynomial of degree p and $x_i \sim \mathcal{N}(0, I_d)$, $i \in [n]$. Then there exists an absolute positive constant C_p depending only on p such that for any $\delta > 0$,*

$$\mathbb{P}\left[|\frac{1}{n}\sum_{i=1}^n g(x_i) - \mathbb{E}[g(x)]| \geqslant \delta\frac{1}{\sqrt{n}}\sqrt{\mathrm{Var}(g(x))}\right] \leqslant 2\exp\left(-C_p \min\left(\delta^2, \delta^{2/p}\right)\right)$$

E.4 Uniform Generalization Bounds

Definition 5 (Rademacher complexity)**.** The empirical Rademacher complexity of a function class $\mathcal{F}$ on finite samples is defined as

$$\widehat{\mathrm{Rad}}_n(\mathcal{F}) = \mathbb{E}_\xi\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \xi_i f(X_i)\right] \tag{23}$$

where $\xi_1, \xi_2, \ldots, \xi_n$ are i.i.d. Rademacher random variables: $\mathbb{P}(\xi_i = 1) = \mathbb{P}(\xi_i = -1) = \frac{1}{2}$. Let $\mathrm{Rad}_n(\mathcal{F}) = \mathbb{E}[\widehat{\mathrm{Rad}}(\mathcal{F})]$ be the population Rademacher complexity.

Then we recall the uniform law of large number via Rademacher complexity, which can be found in Wainwright (2019, Theorem 4.10).

Lemma 33. *Assume that f ranges in $[0, R]$ for all $f \in \mathcal{F}$. For any $n \geqslant 1$, for any $\delta \in (0, 1)$, w.p. at least $1 - \delta$ over the choice of the i.i.d. training set $S = \{X_1, \ldots, X_n\}$, we have*

$$\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^n f\left(X_i\right) - \mathbb{E}f(X)\right| \leqslant 2\,\mathrm{Rad}_n(\mathcal{F}) + R\sqrt{\frac{\log(4/\delta)}{n}} \tag{24}$$

Then we recall the contraction Lemma in Vershynin (2018, Exercise 6.7.7) to compute Rademacher complexity.

Lemma 34 (Contraction Lemma)**.** *Let $\varphi_i : \mathbb{R} \mapsto \mathbb{R}$ with $i = 1, \ldots, n$ be β-Lispchitz continuous. Then,*

$$\frac{1}{n}\mathbb{E}_\xi \sup_{f\in\mathcal{F}}\sum_{i=1}^n \xi_i \varphi_i \circ f\left(x_i\right) \leqslant \beta\widehat{\mathrm{Rad}}_n(\mathcal{F})$$

Next, we estimate the Rademacher complexity for random feature models. Denote $g_{u,s,V}(x) = u^\top \sigma\left(\frac{Vx+s}{\sqrt{2}}\right) = \sum_{i=1}^m u_i \sigma\left(\frac{v_i^\top x + s_i}{\sqrt{2}}\right)$ with v_i i.i.d. sampled from the uniform distribution on the unit sphere, and s_i i.i.d. $\mathcal{N}(0,1)$ generated. Here $\sigma(\cdot)$ is a k degree polynomial with $\mathcal{O}(1)$ coefficients. Denote our kernel function class $\mathcal{G}$ as

$$\mathcal{G} := \{g_{u,s,V} : \|u\| \leqslant M_u\}$$

Then we have the following lemma for the Rademacher complexity of $\mathcal{G}$.

Lemma 35. *With probability at least $1 - 2e^{-\Theta(1)m^{1/2k}}$, we have the following estimation for the Rademacher complexity of our function class $\mathcal{G}$*

$$\mathrm{Rad}_n(\mathcal{G}) \lesssim M_u \sqrt{\frac{m}{n}}$$

Proof.

$$\begin{aligned}
\mathrm{Rad}_n(\mathcal{G}) &= \mathbb{E}_{x,\xi}\left[\sup_{g_\theta \in \mathcal{G}} \frac{1}{n}\sum_{i=1}^n \xi_i u^\top \sigma\left(\frac{Vx_i+s}{\sqrt{2}}\right)\right] \\
&= \frac{1}{n}\,\mathbb{E}_{x,\xi}\left[\sup_{g_\theta \in \mathcal{G}} u^\top \left(\sum_{i=1}^n \xi_i \sigma\left(\frac{Vx_i+s}{\sqrt{2}}\right)\right)\right] \\
&\leqslant \frac{M_u}{n}\,\mathbb{E}_{x,\xi}\left[\left\|\sum_{i=1}^n \xi_i \sigma\left(\frac{Vx_i+s}{\sqrt{2}}\right)\right\|_2\right] \\
&\leqslant \frac{M_u}{n}\sqrt{\mathbb{E}_{x,\xi}\left[\left\|\sum_{i=1}^n \xi_i \sigma\left(\frac{Vx_i+s}{\sqrt{2}}\right)\right\|_2^2\right]} \\
&= \frac{M_u}{n}\sqrt{\mathbb{E}_x\left[\sum_{j=1}^m \mathrm{Var}_\xi\left(\sum_{i=1}^n \xi_i \sigma\left(\frac{v_j^\top x_i + s_j}{\sqrt{2}}\right)\right)\right]} \\
&= \frac{M_u}{\sqrt{n}}\sqrt{\mathbb{E}_x\left[\sum_{j=1}^m \sigma\left(\frac{v_j^\top x + s_j}{\sqrt{2}}\right)^2\right]} \lesssim M_u\sqrt{m}\sqrt{\frac{\frac{1}{m}\sum_{j=1}^m (1+s_j^{2k})}{n}}
\end{aligned} \tag{25}$$

By Corollary 3, we can concentrate $\frac{1}{m}\sum_{j=1}^m (1+s_j^{2k})$ and get

$$\frac{1}{m}\sum_{j=1}^m (1+s_j^{2k}) \lesssim 1$$

with probability at least $1 - 2e^{-\Theta(1)m^{1/2k}}$. Plug that in and we get our final bound. □

E.5 Convex Optimization

Denote $f(x)$ as a C^1 function defined in $\mathbb{R}^d$. Assume that

- There exists $m > 0$ such that $f(x) - \frac{m}{2}\|x\|^2$ is convex.
- $\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x-y\|$.

The following result is standard and can be found in most convex optimization textbooks like Boyd & Vandenberghe (2004).

Lemma 36. *There exists a unique x^* such that $f(x^*) = \inf_x f(x)$. And if we start at the point x^0 and do gradient descent with learning rate η, if $\eta \leqslant \frac{1}{m+L}$, then we will get*

$$\left\|x^k - x^*\right\|^2 \leqslant c^k \left\|x^0 - x^*\right\|^2$$

where $c = 1 - \eta\frac{2mL}{m+L}$.

E.6 UNIVARIATE APPROXIMATION

In this subsection, we use $\sigma(z)$ to denote $\mathrm{ReLU}(z)$ and set $A \geqslant 1$.

Lemma 37. *Let $a \sim \mathrm{Unif}(\{-1,1\})$ and let b have density $\mu_b(t)$. Then there exists $v(a,b)$ supported on $\{-1,1\} \times [A,2A]$ such that for any $|x| \leqslant A$,*

$$\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] = 1 \quad \textit{and} \quad \sup_{a,b} |v(a,b)| \leqslant \frac{1}{\int_A^{2A} t\mu_b(t)dt}$$

Proof. Let $v(a,b) = c\mathbf{1}_{b\in[A,2A]}$ where $c = \frac{1}{\int_A^{2A} t\mu_b(t)dt}$. Then for $|x| \leqslant A$,

$$\begin{aligned}
\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] &= c\int_A^{2A} \frac{1}{2}[\sigma(x+t)+\sigma(-x+t)]\mu_b(t)dt \\
&= c\int_A^{2A} t\mu_b(t)dt \\
&= 1
\end{aligned}$$

□

Lemma 38. *Let $a \sim \mathrm{Unif}(\{-1,1\})$ and let b have density $\mu_b(t)$. Then there exists $v(a,b)$ supported on $\{-1,1\} \times [A,2A]$ such that for any $|x| \leqslant A$,*

$$\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] = x \quad \textit{and} \quad \sup_{a,b} |v(a,b)| \leqslant \frac{1}{\int_A^{2A} \mu_b(t)db}$$

Proof. Let $v(a,b) = ca\mathbf{1}_{b\in[A,2A]}$ where $c = \frac{1}{\int_A^{2A} \mu_b(t)dt}$. Then for $|x| \leqslant A$,

$$\begin{aligned}
\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] &= c\int_A^{2A} \frac{1}{2}[\sigma(x+t)-\sigma(-x+t)]\mu_b(t)dt \\
&= cx\int_A^{2A} \mu_b(t)dt \\
&= x
\end{aligned}$$

□

Lemma 39. *Let $a \sim \mathrm{Unif}(\{-1,1\})$ and let b have density $\mu_b(t)$. Let $f : \mathbb{R} \to \mathbb{R}$ be any C^2 function. Then there exists $v(a,b)$ supported on $\{-1,1\} \times [0,2A]$ such that for any $|x| \leqslant A$,*

$$\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] = f(x)$$

and

$$\sup_{a,b} |v(a,b)| = \mathcal{O}\left(\sup_{x\in[-A,A],k=0,1,2} \left|f^{(k)}(x)\right| \left(\frac{1}{\int_A^{2A} \mu_b(t)dt} + \frac{1}{\inf_{t\in[0,A]} \mu_b(t)}\right)\right)$$

Proof. First consider $v(a,b) = \frac{\mathbf{1}_{b\in[0,A]}}{\mu_b(t)} 2f''(-ab)$. Then when $x \geqslant 0$ we have the following equation by integration by parts:

$$\begin{aligned}
&\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] \\
&= \int_0^A [f''(-t)\sigma(x+t) + f''(t)\sigma(-x+t)]\,dt \\
&= x(f'(0)-f'(-A)) - Af'(-A) + f(0) - f(-A) + Af'(A) - f(A) + f(x) - xf'(A) \\
&= f(x) + C_1 + C_2 x
\end{aligned}$$

where $C_1 = -Af'(-A) + f(0) - f(-A) + Af'(A) - f(A)$ and $C_2 = f'(0) - f'(-A) - f'(A)$. In addition when $x < 0$,

$$\begin{aligned}
&\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] \\
&= \int_0^A [f''(-t)\sigma(x+t) + f''(t)\sigma(-x+t)]\, dt \\
&= x(f'(0) - f'(-A)) - Af'(-A) + f(0) - f(-A) + Af'(A) - f(A) + f(x) - xf'(A) \\
&= f(x) + C_1 + C_2 x
\end{aligned}$$

so this equality is true for all x. We can use the previous two lemmas to subtract the $C_1 + C_2 x$ term. That is to say, we can set

$$v(a,b) := -C_1 \frac{1}{\int_A^{2A} t\mu_b(t)dt} \mathbf{1}_{b\in[A,2A]} - C_2 \frac{a}{\int_A^{2A} \mu_b(t)dt} \mathbf{1}_{b\in[A,2A]} + \frac{1}{\mu_b(t)} \mathbf{1}_{b\in[0,A]} 2f''(-ab)$$

in order to have $\mathbb{E}_{a,b}[v(a,b)\sigma(ax+b)] = f(x)$ for any $|x| \leqslant A$. In this case, we have

$$\sup_{a,b} |v(a,b)| = \mathcal{O}\left(\sup_{x\in[-A,A],k=0,1,2} \left| f^{(k)}(x) \right| \left(\frac{1}{\int_A^{2A} \mu_b(t)dt} + \frac{1}{\inf_{t\in[0,A]} \mu_b(t)} \right) \right)$$

□

Remark 7. When f is a polynomial and $\mu_b(t)$ has a heavy tail, $\sup_{a,b} |v(a,b)|$ will only depend on A polynomially. More concretely, consider the case $f(z) = \sum_{0\leqslant i\leqslant q} c_i z^i$ where $\sup_i |c_i| = \mathcal{O}(1)$. In this case, we have

$$\sup_{x\in[-A,A],k=0,1,2} \left| f^{(k)}(x) \right| = \mathcal{O}(A^q)$$

Furthermore, since we have assumed $\mu_b(t) \gtrsim (|t|+1)^{-p}$, we have

$$\left(\frac{1}{\int_A^{2A} \mu_b(t)dt} + \frac{1}{\inf_{t\in[0,A]} \mu_b(t)} \right) = \mathcal{O}(A^p) \text{ and } \sup_{a,b} |v(a,b)| = \mathcal{O}(A^{p+q})$$