ALGORITHMIC SEPARATION BETWEEN CONSTANT-DEPTH AND LOGARITHMIC-DEPTH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the empirical advantages of deep networks over shallow ones, theoretical depth separations largely concern approximation power, while algorithmic results are mostly limited to comparisons between two- and three-layer networks. In this work, we prove the first algorithmic separation between constant-depth and logarithmic-depth networks.

Specifically, we identify a class of Boolean functions with hierarchically structured Fourier spectra that logarithmic-depth networks can learn efficiently using layerwise coordinate descent by reconstructing the spectra hierarchically and adaptively. We also exhibit a subclass for which every constant-depth, polynomial-width network with sufficiently regular activations and controlled spectral norms must incur constant L^2 approximation error under the uniform distribution over the hypercube.

1 INTRODUCTION

One of the central questions in deep learning theory is how to formalize the advantage of deep models over shallow ones. Beginning with the works of Eldan & Shamir (2016); Telgarsky (2016), a long line of research has sought to separate the approximation power of deep and shallow networks (Daniely, 2017; Safran & Shamir, 2017; Venturi et al., 2022; Safran et al., 2025). These works exhibit functions that deep networks can approximate efficiently but shallow networks cannot.

However, due to the nonconvex nature of deep network training, being able to approximate a function does not imply that a network trained with gradient-based methods can efficiently find that solution. Therefore, it is unclear whether this approximation-based separation can properly explain the empirical advantage of deep networks. This motivates the pursuit of *algorithmic* separation between deep and shallow networks. Namely, the goal is to find function classes that cannot be efficiently learned, or better, approximated by a shallow network, but can be efficiently learned by a deep network.

In contrast to approximation-based separation, algorithmic-separation results are much rarer, and most concern models with two or three layers (Safran & Lee, 2022; Ren et al., 2023; Nichani et al., 2023). In this work, we prove the first algorithmic separation between constant-depth and logarithmic-depth networks. Our target class consists of hierarchical staircase functions — Boolean functions whose Fourier spectra have a hierarchical structure, in the sense that high-degree terms can be constructed by combining a constant number of low-degree terms.

We formally define hierarchical staircase functions in Definition 1.2. Here, as a running example and prototype of hierarchical staircase functions, we introduce the base deep quadratic function (cf. Appendix C of Ren et al. (2026)). Suppose that $d = 2^{L_*}$ for some $L_* \in \mathbb{N}_+$. The base depth-L_* deep quadratic function is defined as $f_*(\boldsymbol{x}) = L_*^{-1/2} \sum_{l=1}^{L_*} 2^{-(L_*-l)/2} f_l(\boldsymbol{x})$, where $f_1(\boldsymbol{x}) = x_1x_2 + x_3x_4 + \cdots + x_{d-1}x_d$, $f_2(\boldsymbol{x}) = x_1 \cdots x_4 + \cdots + x_{d-3} \cdots x_d$, $\ldots$, $f_{L_*}(\boldsymbol{x}) = x_1 \cdots x_d$. General deep quadratic functions can be obtained by permuting the coordinates of this f_* in each layer. See Definition 4.1 for the formal definition.

It is clear that for any deep quadratic function f_*, the total mass of terms with degree at least $\sqrt{d} = 2^{L_*/2}$ is at least $1/3$. Therefore, to learn f_* to $o(1)$-accuracy, the learner network must capture those

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

high-degree terms. The inability to do so is what we show in the lower bound against constant-depth networks.

On the other hand, to learn f_* using an L-layer network, it suffices to learn the features $(x_1x_2, \dots, x_{d-1}x_d)$ in the first layer, combine them into the features $(x_1 \cdots x_4, \dots, x_{d-3} \cdots x_d)$ in the second layer using a quadratic mapping, and continue recursively. In other words, a deep network can learn the features *hierarchically* and *adaptively*.

Theorem 1.1 (Informal version of Theorems 2.1, 3.1 and 4.1)**.** *Suppose that* $d = 2^{L_*}$.

- **(Lower bound against shallow networks)** *No constant-depth fully connected network with sufficiently regular activations and* d^{c_0}*-spectrally bounded weights can approximate any depth-*L_* *deep quadratic function, where* $c_0 > 0$ *is a sufficiently small constant.*
- **(Upper bound for deep networks)** *An* $(L_* + 1)$*-layer network, trained by layerwise one-pass coordinate descent, can efficiently learn the class of hierarchical staircase functions with bounded* L^2*-norm, which includes all depth-*L_* *deep quadratic functions.*

We remark that our approximation-based lower bound is *stronger* than optimization-based ones, as it rules out any algorithm that outputs a constant-depth network satisfying the required conditions.

1.1 Our contributions

Polynomial approximation and lower bound. Let $f_* : \{\pm 1\}^d \to \mathbb{R}$ be such that the total Fourier mass of terms with degree at least $d^{1/2}$ is at least $1/2$. Then, by the orthogonality of Boolean monomials under $\mathrm{Unif}(\{\pm 1\}^d)$, no polynomial of degree at most $o(d^{1/2})$ can approximate f_* to L^2-error smaller than $1/2$. We prove the following polynomial approximation lemma for constant-depth networks, which may be of independent interest, and then derive the lower bound from it.

Lemma 1.1 (Informal version of Lemma 2.2)**.** *Any constant-depth fully connected network with sufficiently regular activations and* C_W*-spectrally bounded weights can be approximated by a polynomial of degree* $(C_W \log d)^{O(1)}$.

Hierarchical learning and upper bound. For our upper bound, we show that a depth-L network can efficiently learn depth-L hierarchical staircase functions, which are formally defined below.

Definition 1.2 (Hierarchical staircase function)**.** *Let* $f : \{\pm 1\}^d \to \mathbb{R}$ *be a Boolean function and* $\hat{f} : 2^{[d]} \to \mathbb{R}$ *be its Fourier transform. Let* $\mathcal{S}_f := \left\{S \subset [d] : \hat{f}(S) \neq 0\right\}$ *denote the (Fourier) spectrum of* f*. For* $L, D \in \mathbb{N}$*, we say that* f *is a hierarchical staircase function of depth* L *and local degree* D *if* $\mathcal{S}_f \subset \mathcal{K}_L$ *with* $(\mathcal{K}_l)_{l \in \mathbb{N}} \subset 2^{[d]}$ *defined inductively by*

$$\mathcal{K}_1 = \{\varnothing, \{1\}, \dots, \{d\}\}, \quad \mathcal{K}_{l+1} = \mathcal{K}_l \cup \left(\mathcal{K}_l^{\oplus D} \cap \mathcal{S}_f\right), \tag{1}$$

where $\mathcal{K}^{\oplus D} := \{K_1 \oplus \cdots \oplus K_D : K_1, \dots, K_D \in \mathcal{K}\}$ *with* $\oplus$ *denoting the symmetric difference. In addition, we call* $|\mathcal{S}_f|$ *the width of* f *and* $\min_{S \in \mathcal{S}_f} |\hat{f}(S)|$ *the signal strength of* f.

For any $L, D, m \in \mathbb{N}_+$ *and* $\kappa > 0$*, we use* $\mathrm{HS}_{L,D,m,\kappa}$ *to denote the collection of hierarchical staircase functions with depth at most* L*, local degree at most* D*, width at most* m *and signal strength at least* κ.

It is clear that $\mathcal{K}_l \subset \mathcal{K}_{l+1} \subset \mathcal{S}_f \cup \{\varnothing, \{1\}, \dots, \{d\}\}$. In words, (1) means that at each step, we add to $\mathcal{K}_l$ the sets in $\mathcal{S}_f$ that can be constructed by combining D existing terms in $\mathcal{K}_l$. Thus, if every higher-order term in $\mathcal{S}_f$ can be constructed from lower-order terms in $\mathcal{S}_f$, possibly together with terms of degree at most D, then $\mathcal{K}_l$ eventually contains $\mathcal{S}_f$. We initialize $\mathcal{K}_1 = \{\varnothing, \{1\}, \dots, \{d\}\}$ to include staircase functions such as $x_1 + x_1x_2 + \cdots + x_1 \cdots x_L$ (Abbe et al., 2021; 2022; 2023).

Also note that to learn hierarchical staircase functions, the model needs to learn the monomials not only hierarchically but also adaptively. Since there are potentially exponentially many monomials, the model cannot simply check if each of them belongs to $\mathcal{S}_f$ even when $|\mathcal{S}_f| = \mathrm{poly}(d)$.

Theorem 1.2 (Informal version of Theorem 3.1)**.** *For any* $L, D, m \in \mathbb{N}_+, \kappa \in \mathbb{R}_+$ *with* $D = O(1)$*, there is a depth-*L *network that, when trained with empirical coordinate descent, can learn the class* $\mathrm{HS}_{L,D,m,\kappa}$ *of hierarchical staircase functions. Moreover, the model size, sample complexity, and running time are all polynomials in* $d, L, m, 1/\kappa$.

1.2 Related work

Depth separation and lower bounds. Classical depth-separation results show that functions efficiently represented by deep networks may require much larger shallow networks (Eldan & Shamir, 2016; Telgarsky, 2016; Daniely, 2017; Safran & Shamir, 2017; Venturi et al., 2022; Wang & Wu, 2024; Safran et al., 2025). These results primarily concern approximation, while the comparatively few algorithmic separations focus on networks with two or three layers (Safran & Lee, 2022; Ren et al., 2023; Nichani et al., 2023). Our lower bound rules out the approximation of deep quadratic functions by constant-depth, polynomial-width networks. Combined with our efficient learning guarantee, it yields an algorithmic separation between constant and logarithmic depth. Technically, most prior lower bounds rely on oscillation counting or special structural properties of two-layer networks. In contrast, our lower bound is based on approximating constant-depth networks by Chebyshev polynomials. The depth-2 lower bound in Daniely (2017) also uses the failure of low-degree polynomial approximation, but the argument relies heavily on harmonic analysis on the sphere and cannot be easily extended to more than two layers. Recently, Bruna (2026) established an algorithmic separation beyond the two-vs-three setting using a different model and approach. However, his lower bound still concerns only two-layer networks.

Hierarchical learning and upper bounds. Our upper bound is related to work showing that neural networks can progressively construct hierarchical features. This includes staircase and merged-staircase functions (Abbe et al., 2021; 2022; 2023), two-level compositional targets learned by three-layer networks (Allen-Zhu & Li, 2019; Nichani et al., 2023; Wang et al., 2024; Fu et al., 2025; Dandi et al., 2025; Tabanelli et al., 2026), and guarantees for genuinely deep hierarchies (Allen-Zhu & Li, 2023; Daniely, 2026; Ren et al., 2026; Dandi et al., 2026; Bruna, 2026). Our hierarchical staircase condition allows each new Fourier feature to be composed from up to D previously constructed features, so feature degree can grow exponentially with depth. This function class is much larger than the staircase functions considered in Abbe et al. (2021) and does not require the number of relevant coordinates to be constant as in Abbe et al. (2022; 2023). See Appendix E.1 for further discussion. In addition, unlike the preceding hierarchical-learning results, our learning guarantee is paired with a lower bound showing that constant-depth networks (with potentially more than two layers) cannot learn the subclass of deep quadratic functions.

Our upper bound proof is built on the shallow-to-deep chaining principle (cf. Informal Principle 2 of Ren et al. (2026)). See Section 3 for more discussion. Our proof substantially extends the discussion in Appendix C of Ren et al. (2026), which is informal and additionally assumes that features at the same level have disjoint supports. In this work, we rigorously handle overlapping supports and control duplicated feature through finite-sample coordinate descent. More recently, Dandi et al. (2026) studies a similar mechanism. Our settings are quite different, and the resulting upper bounds are not directly comparable. Neither of these two works provides an accompanying lower bound.

1.3 Notation and conventions

For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_p$ denotes its ℓ_p-norm. When $p = 2$, we will often omit the subscript and write $\|\boldsymbol{x}\|$. For a matrix $\boldsymbol{W}$, $\|\boldsymbol{W}\|_{\mathrm{OP}}$ denotes its operator norm. For $a, b, \delta \in \mathbb{R}$, $a = b \pm \delta$ means $|a - b| \le \delta$. We define $[N] = \{1, \dots, N\}$ and $[N]_0 = \{0\} \cup [N]$ for any $N \in \mathbb{N}_{>0}$. In addition, $\binom{[N]}{\le D}$ denotes the collection of subsets of $[N]$ of size at most D. For any $S \subset [d]$, let $\chi_S : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \mapsto \prod_{i \in S} x_i$ denote the multilinear monomial associated with S. The functions $(\chi_S)_{S \subset [d]}$ form an orthonormal basis of real-valued Boolean functions with respect to the uniform distribution over $\{\pm 1\}^d$. We will only use this basic property in this paper. For more background on Boolean analysis, see O'Donnell (2014). Throughout this paper, the input distribution is always the uniform distribution over $\{\pm 1\}^d$ with $d \gg 1$.

1.4 Outline

The rest of this paper is organized as follows. We formally state our lower and upper bounds and sketch their proofs in Sections 2 and 3, respectively. Then in Section 4, we instantiate the above bounds on deep quadratic functions and prove our depth-separation theorem. We also complement our theoretical results with experiments on ReLU networks in Section 4. We conclude in Section 5. See the appendix for the table of contents.

2 LOWER BOUND AGAINST CONSTANT-DEPTH NETWORKS

In this section, we describe our lower bound against constant-depth networks and sketch its proof. The complete proof can be found in Appendix B.

2.1 SETUP AND MAIN RESULT

Let $\boldsymbol{x} \in \{\pm 1\}^d$ denote the input and set $\boldsymbol{x}^{(1)} := \boldsymbol{x}$ and $d_1 := d$. Then, we inductively define our learner network as

$$\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) = \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x})\right), \quad \forall l \in [L], \quad f_L(\boldsymbol{x}) := \left\langle \boldsymbol{a}, \boldsymbol{x}^{(L+1)}(\boldsymbol{x})\right\rangle, \tag{2}$$

where $\boldsymbol{W}^{(l)} = (\boldsymbol{w}_k^{(l)})_{k\in[d_{l+1}]}^\top \in \mathbb{R}^{d_{l+1}\times d_l}$ and $\boldsymbol{a} \in \mathbb{R}^{d_{L+1}}$ are the weights of the network, and $\boldsymbol{\phi}_l : \mathbb{R}^{d_{l+1}} \to \mathbb{R}^{d_{l+1}}$ are activation functions. We assume $\boldsymbol{\phi}_l$ are applied entrywise in the sense that $\boldsymbol{\phi}_l(\boldsymbol{z}) = (\phi_{l,k}(z_k))_{k\in[d_{l+1}]}$, where $\phi_{l,k} : \mathbb{R} \to \mathbb{R}$. Allowing coordinate-dependent activation functions lets us absorb the bias terms into the activations and use more compact notation. Also, we use $\boldsymbol{\phi}_l'(\boldsymbol{z}) = (\phi_{l,k}'(z_k))_{k\in[d_{l+1}]}$ to denote the entrywise derivative of $\boldsymbol{\phi}_l$. We will assume the following about the network.

Assumption 2.1. *Consider the network defined in (2). Put $d_0 = d$ and $d_{\max} = \max_{l\in[L+1]_0} d_l$. We assume that there are $C_W, C_\phi \geq 1$ and universal constants $c_\phi > 0$ and $m_\phi \geq 0$ such that the following conditions hold:*

(a) L is a universal constant; $d_{\max} \leq d^{O(1)}$; and $C_W, C_\phi \leq d^{c_0}$ with $c_0 < 2^{-2L-7}/L$.

(b) $\left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}} \leq C_W$ for all $l \in [L]$ and $\|\boldsymbol{a}\| \leq 1$.

(c) For every $l \in [L]$ and $k \in [d_{l+1}]$, $|\phi_{l,k}(0)| \vee \left\|\phi_{l,k}'\right\|_{L^\infty} \vee \left\|\phi_{l,k}''\right\|_{L^\infty} \leq C_\phi$.

(d) For every $l \in [L]$ and $k \in [d_{l+1}]$, $\phi_{l,k}$ can be analytically continued to the strip $\{z \in \mathbb{C} : |\mathrm{Im}\, z| < c_\phi\}$ and the continued $\phi_{l,k}$ satisfies the following strip-growth condition:

$$|\phi_{l,k}(a+ib)| \leq C_\phi(1+|a|)^{m_\phi}, \quad a, b \in \mathbb{R},\ |b| \leq c_\phi/2.$$

Remark. Condition (b) prevents the norm of intermediate representations from blowing up too quickly. Under standard schemes such as Xavier initialization Glorot & Bengio (2010) and He initialization He et al. (2015), this condition holds with $C_W = O(1)$ at initialization with high probability when adjacent layer widths are of the same order. This condition is also compatible with the μP and feature learning theory (Yang & Hu, 2021; Yang et al., 2024).

Condition (d) is used to show that ϕ can be approximated uniformly by Chebyshev polynomials on finite intervals (cf. Lemma A.6). While this condition excludes ReLU activation, we verify this condition for common smooth activation functions, such as $\tanh$, sigmoid, and smooth variants of ReLU such as softplus, and GELU (cf. Lemma A.7). We also provide experimental results on ReLU networks in Figure 1. ♣

As noted in the introduction, our lower bound is based on polynomial approximation of the network. The following is our main polynomial approximation lemma, which may be of independent interest.

Lemma 2.2. *Suppose that Assumption 2.1 holds. Then, for any $\varepsilon_* \geq d^{-O(1)}$, there exists a polynomial $\tilde{f}_L : \{\pm 1\}^d \to \mathbb{R}$ such that $\left\|f_L - \tilde{f}_L\right\|_{L^2} \leq \varepsilon_*$ and*

$$\deg \tilde{f}_L \lesssim \left((C_\phi C_W^2)^{2^{L+3}}(\log d)^{2L+2}\right)^{L(2^L+1)} \ll d^{1/2}.$$

The following theorem is a direct corollary of Lemma 2.2.

Theorem 2.1 (Lower bound against constant-depth networks)**.** *Let $f_* : \{\pm 1\}^d \to \mathbb{R}$ be a Boolean function whose Fourier coefficients satisfy $\sum_{|S|\geq d^{1/2}} \hat{f}_*^2(S) \geq 1/3$. Then, for any network f satisfying Assumption 2.1, we have $\|f_L - f_*\|_{L^2(\mathrm{Unif}(\{\pm 1\}^d))} \geq 1/4$ for all sufficiently large d.*

2.2 PROOF SKETCH

For ease of presentation, we assume in this subsection that C_W and C_ϕ are also universal constants. The general case is rigorously handled in Appendix B. Here, we call a polynomial low degree if its degree is at most $\operatorname{poly}\log d$, and we use high degree to mean degree at least $d^{1/2}$. As we have mentioned earlier, it suffices to show that f_L can be approximated by a low-degree polynomial.

To this end, we replace each non-polynomial activation $\boldsymbol{\phi}_l$ with its polynomial approximation. For each $l \in [L]$, put $\bar{\boldsymbol{z}}^{(l)} := \boldsymbol{W}^{(l)} \mathbb{E}\, \boldsymbol{x}^{(l)} \in \mathbb{R}^{d_{l+1}}$, and let $R_l \geq 1, \varepsilon_\phi > 0$ be parameters to be chosen later. For every $k \in [d_{l+1}]$, let $\tilde{\phi}_{l,k}$ be the degree-p_l Chebyshev approximation to $\phi_{l,k}$ on the interval $I_k^{(l)} := [\bar{z}_k^{(l)} - R_l, \bar{z}_k^{(l)} + R_l]$, where the degree p_l is chosen so that the $\|\cdot\|_{L^\infty}$-difference between $\phi_{l,k}$ and $\tilde{\phi}_{l,k}$ over $I_k^{(l)}$ is bounded by ε_ϕ. See Appendix A for background on Chebyshev polynomials and approximation. Then, define

$$\tilde{\boldsymbol{x}}^{(l+1)}(\boldsymbol{x}) = \tilde{\boldsymbol{\phi}}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right), \quad \forall l \in [L], \quad \tilde{f}_L(\boldsymbol{x}) := \left\langle \boldsymbol{a}, \tilde{\boldsymbol{x}}^{(L+1)}(\boldsymbol{x})\right\rangle, \tag{3}$$

where $\tilde{\boldsymbol{x}}^{(1)} = \boldsymbol{x}^{(1)} = \boldsymbol{x}$. It is clear that $\tilde{f}_L$ is a polynomial of degree $p_1 \cdots p_L$. Our goal is to show that we can choose $(R_l)_l, \varepsilon_\phi$ properly, so that $\left\|f_L - \tilde{f}_L\right\|_{L^2}$ is small and the total degree $p_1 \cdots p_L$ is much smaller than $d^{1/2}$.

Consider an intermediate layer $l \in [L]$ and fix $q \geq 1$. For the approximation error at layer l, we have

$$\begin{aligned}\left\|\boldsymbol{x}^{(l+1)} - \tilde{\boldsymbol{x}}^{(l+1)}\right\|_{L^q} &= \left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}\right) - \tilde{\boldsymbol{\phi}}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}\right)\right\|_{L^q} \\ &\leq \left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}\right) - \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}\right)\right\|_{L^q} + \left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}\right) - \tilde{\boldsymbol{\phi}}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}\right)\right\|_{L^q}.\end{aligned}$$

By the Lipschitzness of $\boldsymbol{\phi}_l$ and the operator-norm bound on $\boldsymbol{W}^{(l)}$, we can bound the first error term using $\left\|\boldsymbol{x}^{(l)} - \tilde{\boldsymbol{x}}^{(l)}\right\|_{L^q}$. Hence, to obtain a recurrence relation that controls the approximation error, it suffices to analyze the second error term.

One can show that, for every $l \in [L]$, the exact intermediate representation $\boldsymbol{x}^{(l)}$ is subgaussian after centering (cf. Lemma B.4). Consequently, the corresponding preactivations lie in intervals of length at most $\operatorname{poly}\log(d)$ with high probability. We seek an analogous statement for the approximate preactivations $(\boldsymbol{w}_k^{(l)} \cdot \tilde{\boldsymbol{x}}^{(l)})_k$. Because $\tilde{\boldsymbol{\phi}}_l$ approximates $\boldsymbol{\phi}_l$ over the intervals $(I_k^{(l)})_k$, a natural strategy is to choose intervals of length $\operatorname{poly}\log d$, show that the approximate preactivations lie in them with high probability, and control the contribution from the complement.

Unfortunately, the above strategy cannot be implemented in a straightforward global way. To approximate $\boldsymbol{\phi}_l$ on larger intervals using polynomials, generally we need polynomials of higher degrees (cf. Appendix A and Lemma A.5). Consequently, the contribution outside the intervals can also grow.

To see why this prevents a direct global argument, suppose that the degree and interval length at every layer are p and R, respectively. By Lemma A.5, this requires choosing $p \gtrsim R \log R \geq R$. Hence, the total degree will be at least R^L and the R^L-th moment of a subgaussian variable can be as large as R^{LR^L}. On the other hand, the bound on the failure probability is at best $e^{-\Theta(R^2)}$. Hence, in order for $e^{-\Theta(R^2)} R^{LR^L} \ll 1$ to hold, we would need $LR^L \ll R^2$, which is impossible when $L \geq 2$.

To fix the above issue, we will use a layerwise argument. Let $\mathcal{G}_l$ denote the event that all preactivations at layer l lie in the corresponding $I_k^{(l)}$. Then, let $\mathcal{B}_l$ denote the event that the first failure happens at layer l. That is, $\mathcal{B}_l = \mathcal{G}_1 \cap \cdots \cap \mathcal{G}_{l-1} \cap \mathcal{G}_l^c$. Now, consider layer l and event $\mathcal{B}_l$. Since the preactivations in earlier layers all lie in their corresponding approximation interval, one can show that the layer-l inputs are subgaussian (cf. Lemma B.7). As a function of the layer-l input, $\tilde{f}_L$ is a polynomial of degree $D_l := \prod_{k=l}^L p_k = p_l D_{l+1}$. Hence, typical size of the final output is of order at most $d^{D_l} = d^{p_l D_{l+1}} \approx d^{R_l (\log R_l) D_{l+1}}$. Meanwhile, the failure probability at this layer is roughly $e^{-\Theta(R_l^2)}$. Therefore, the constraint at this layer is $R_l^2 \gg R_l (\log R_l)(\log d) D_{l+1}$. Hence, when $R_l \leq d$, it suffices to require $R_l \gg (\log d)^2 D_{l+1}$. Note that $D_{L+1} = 1$ and D_{l+1} does not

depend on R_l. Hence, we can solve the above constraints backward, from layer L to layer 1. At each layer, the degree roughly doubles. Because the number of layers is constant, both the degree and the interval length remain polylogarithmic.

3 Upper bound for logarithmic-depth networks

In this section, we first describe our learner architecture and training algorithm. Then, we formally state our upper bound for logarithmic-depth networks and sketch the proof. The complete proof can be found in Appendix C.

Throughout this section, $f_* : \{\pm 1\}^d \to \mathbb{R}$ will denote our target function and $\mathcal{S}_*$ its Fourier spectrum. We assume $f_* \in \mathrm{HS}_{L,D,m,\kappa}$. That is, f_* is a hierarchical staircase function with depth L, local degree D, width m, and signal strength κ. In addition, we assume $\|f_*\|_{L^2} \le 1$.

Recall from the introduction that a hierarchical staircase function f_* is defined to have a Fourier spectrum $\mathcal{S}_*$ that can be recovered through the recurrence relation:

$$\mathcal{K}_1 = \{\varnothing, \{1\}, \dots, \{d\}\}, \quad \mathcal{K}_{l+1} = \mathcal{K}_l \cup \left(\mathcal{K}_l^{\oplus D} \cap \mathcal{S}_*\right).$$

This definition suggests the following conceptual algorithm. At step l, we first form all products of at most D learned features in $\mathcal{K}_l$. Instead of checking all high-degree features, we test only these candidates for membership in $\mathcal{S}_*$ and discard those outside $\mathcal{S}_*$. We then add the remaining features to $\mathcal{K}_l$. The use of sets ensures we keep exactly one copy of each feature. The above pruning and deduplication operations ensure that at each step, we only check and store polynomially many features.

In the following (and Appendix C), we show that the above conceptual algorithm can be implemented by training a deep network using layerwise coordinate descent.

3.1 Setup and main result

Our learner network is an L-layer network. For $l \in [L]$, let $\boldsymbol{y}^{(l)} \in \mathbb{R}^m$ denote the output of the l-th layer of the network.[1] In addition, we set $\boldsymbol{y}^{(0)} = \mathbf{0} \in \mathbb{R}^m$. Let $M = \sum_{k=0}^{D} \binom{d+m}{k}$ denote the number of subsets of $[d+m]$ of size at most D. Fix an arbitrary ordering of $\binom{[d+m]}{\le D}$, so that we can identify $[M]$ with $\binom{[d+m]}{\le D}$ and index the entries of vectors in $\mathbb{R}^M$ using subsets of $[d+m]$. Define the degree-D polynomial feature mapping $\boldsymbol{\Phi}$ to be

$$\boldsymbol{\Phi} : \mathbb{R}^{d+m} \to \mathbb{R}^M, \quad \boldsymbol{x}' \mapsto \left(\chi_H(\boldsymbol{x}')\right)_{H \subset [d+m], |H| \le D}, \tag{4}$$

where $\chi_H(\boldsymbol{x}') := \prod_{i \in H} x'_i$ is the multilinear monomial associated with the set H. Let $\odot$ and $\circ$ denote entrywise multiplication and vector concatenation, respectively. Then, we define our learner network inductively as

$$\begin{aligned}
&l\text{-th layer input:} && \boldsymbol{x}^{(l)} := \boldsymbol{x} \circ \boldsymbol{y}^{(l-1)} \in \mathbb{R}^{d+m}, \\
&l\text{-th layer output:} && \boldsymbol{y}^{(l)} := \mathtt{Sel}_{\boldsymbol{w}^{(l)}}\left(\boldsymbol{w}^{(l)} \odot \boldsymbol{\Phi}\left(\boldsymbol{x}^{(l)}\right)\right) =: \mathtt{Sel}_{\boldsymbol{w}^{(l)}}\left(\boldsymbol{w}^{(l)} \odot \boldsymbol{z}^{(l)}\right) \in \mathbb{R}^m, \\
&\text{model output:} && f(\boldsymbol{x}) := \left\langle \boldsymbol{w}^{(L)}, \boldsymbol{z}^{(L)} \right\rangle \in \mathbb{R},
\end{aligned} \tag{5}$$

where $l \in [L]$, $(\boldsymbol{w}^{(l)})_{l \in [L]}$ are the trainable weights and $\mathtt{Sel} : \mathbb{R}^M \to \mathbb{R}^m$ is defined as

$$\mathtt{Sel}_{\boldsymbol{w}}(\boldsymbol{w} \odot \boldsymbol{z}) := (z'_j)_{j \in [m]} \quad \text{where} \quad \boldsymbol{z}' := (w_i z_i)_{i \in [M] \,:\, w_i \neq 0} \circ (0, 0, \dots) \in \mathbb{R}^{\mathbb{N}_+}.$$

In NumPy/PyTorch-style notation, $\mathtt{Sel}$ can be defined more compactly as

$$\mathtt{Sel}_{\boldsymbol{w}}(\boldsymbol{w} \odot \boldsymbol{z}) := \mathtt{PadZero}\left((\boldsymbol{w} \odot \boldsymbol{z})[\boldsymbol{w} \neq 0], \mathtt{length} = m\right)[:m].$$

Note that after training, if $w_i^{(l)} = 0$, then $w_i^{(l)} z_i^{(l)}$ is identically zero. In other words, $\mathtt{Sel}$ keeps the first m nonzero coordinates. In addition, we define $f^{(l)}(\boldsymbol{x}) := \left\langle \boldsymbol{w}^{(l)}, \boldsymbol{z}^{(l)} \right\rangle$ for all $l \in [L]$. This can be interpreted as the scalar readout from the l-th layer. We also write $\boldsymbol{x}^{(L+1)} := \boldsymbol{x} \circ \boldsymbol{y}^{(L)}$.

[1] Our analysis is still valid when the intermediate width of the learner is strictly larger than m. We choose the intermediate width to be exactly m to avoid introducing an unnecessary symbol for it.

Algorithm 1 Training algorithm (layerwise one-pass coordinate descent)

Algorithm parameters: Threshold $\lambda_w > 0$, number of samples per layer N, step size η, number of steps per coordinate T.
Initialization: Set $\boldsymbol{w}^{(l)} = 0$ for all $l \in [L]$.
for $l \in [L]$ **do** ▷ *layerwise train each layer*
  Gather N fresh i.i.d. samples and define $\mathcal{L}^{(l)}$ as in (6).
  for $H \in [M]$ **do** ▷ *one-pass coordinate descent*
    Run coordinate descent (7) with respect to $w_H^{(l)}$ for T steps with step size η.
    $w_H^{(l)} \leftarrow 0$ if $|w_H^{(l)}| \le \lambda_w$.
return $(\boldsymbol{w}^{(l)})_{l\in[L]}$.

Now, we describe our training algorithm. The pseudocode is given in Algorithm 1. In short, we layerwise train each layer of (5) using one-pass coordinate descent. Namely, the training procedure consists of L stages. In stage $l \in [L]$, we sample N independent input-output pairs. For notational simplicity, let $\hat{\mathbb{E}}^{(l)}$ denote the empirical expectation over this dataset. We choose the loss for the l-th layer to be

$$\mathcal{L}^{(l)}(\boldsymbol{w}^{(l)}) := \frac{1}{2}\hat{\mathbb{E}}^{(l)}\left(f_*(\boldsymbol{x}) - f^{(l)}(\boldsymbol{x})\right)^2 = \frac{1}{2}\hat{\mathbb{E}}^{(l)}\left(f_*(\boldsymbol{x}) - \left\langle \boldsymbol{w}^{(l)}, \boldsymbol{z}^{(l)}\right\rangle\right)^2. \tag{6}$$

We train the lth layer weight $\boldsymbol{w}^{(l)}$ by sequentially minimizing $\mathcal{L}^{(l)}$ with respect to each coordinate using gradient descent. That is, we initialize $\boldsymbol{w}^{(l)}$ to be 0, and sequentially set each of its coordinates $H \in \binom{[d+m]}{\le D} \cong [M]$ by minimizing the one-dimensional (quadratic) function $w_H^{(l)} \mapsto \mathcal{L}^{(l)}(\boldsymbol{w}^{(l)})$ using gradient/coordinate descent:

$$w_H^{(l)}(t+1) = w_H^{(l)}(t) - \eta \partial_{w_H^{(l)}} \mathcal{L}^{(l)}\left(\boldsymbol{w}_{-H}^{(l)} + w_H^{(l)}(t)\boldsymbol{e}_H\right), \quad t \in [T], \tag{7}$$

where $\eta > 0$ is the step size, T is the number of steps, and $\boldsymbol{w}_{-H}$ denotes the vector obtained by setting the H-th entry of $\boldsymbol{w}$ to be 0. Note that $\boldsymbol{w}_{-H}^{(l)}$ does not change when we update $w_H^{(l)}$.

Theorem 3.1. *Let $\varepsilon_*, \delta_{\mathbb{P}} \in (0, 0.1)$ denote the target accuracy and failure probability, respectively. Let $f_* \in \mathrm{HS}_{L,D,m,\kappa}$ be our target function, where D is a universal constant. Suppose that $\|f_*\|_{L^2} \le 1$ and choose the parameters of Algorithm 1 to be*

$$\lambda_w = 2^{-D-1}\kappa, \quad N \gtrsim \frac{M^2 \log(LM/\delta_{\mathbb{P}})}{\kappa^{2D+2}\left(\varepsilon_*^2/m \wedge \kappa^{2D+2}\right)}, \quad \eta = 2^{-2D}, \quad T \gtrsim \frac{\log\left(m/\varepsilon_* \vee 1/\kappa\right)}{\kappa^{2D}\eta}, \tag{8}$$

for every $l \in [L]$. Then, with probability at least $1 - \delta_{\mathbb{P}}$, Algorithm 1 will output a network f satisfying $\|f_* - f\|_{L^2} \le \varepsilon_*$.

Remark. Note that each layer of our learner network (5) is at most a degree-D polynomial. Hence, when the learner depth is constant, (5) cannot learn the high-degree part of the target. In addition, we show in Appendix D that the trained network can be efficiently compiled into a network matching the architecture of our lower bound.

Also, to keep the presentation clean, we did not optimize the sample and runtime complexity. In particular, one can easily remove the κ^{2D} factors in (8) by normalizing the coordinates of each $\boldsymbol{x}^{(l)}$ to $\{\pm 1, 0\}$. ♣

3.2 Proof Sketch

Suppose that, for every $S \in \mathcal{S}_*$, there exist an index k and a nonzero scalar α_k such that the corresponding entry of $\boldsymbol{z}^{(L)} := \boldsymbol{\Phi}(\boldsymbol{x}^{(L)})$ satisfies $z_k^{(L)}(\boldsymbol{x}) = \alpha_k \chi_S(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{\pm 1\}^d$. Set the corresponding output weight to $w_k^{(L)} = \hat{f}_*(S)/\alpha_k$ for each $S \in \mathcal{S}_*$ and set all remaining output weights to zero. Then $f(\boldsymbol{x}) = f_*(\boldsymbol{x})$. Hence, our goal is to show that after training, $\boldsymbol{z}^{(L)}$ will encode all the sets in the Fourier spectrum $\mathcal{S}_*$.

To this end, we will show by induction on $l \in [L]$ that $\boldsymbol{x}^{(l)}$ contains all features present in $\mathcal{K}_l \setminus \{\varnothing\}$.[2] Formally, we maintain the following induction hypothesis.

Assumption 3.1 (Induction hypothesis on the input)**.** *Fix $l \in [L]$. We assume as an induction hypothesis the following properties of the l-th-layer input $\boldsymbol{x}^{(l)} := (\boldsymbol{x}, \boldsymbol{y}^{(l-1)}) \in \mathbb{R}^{d+m}$.*

(a) For every $k \in [d+m]$, $x_k^{(l)}$, as a function of the input $\boldsymbol{x}$, is either identically zero or nowhere zero. We write $I_l := \left\{k \in [d+m] \,:\, x_k^{(l)} \not\equiv 0\right\}$ for the collection of nonzero coordinates.

(b) There exist nonzero scalars $(\alpha_k^{(l)})_{k \in I_l}$ and subsets $(S_k^{(l)})_{k \in I_l}$ of $[d]$ such that $x_k^{(l)} = \alpha_k^{(l)} \chi_{S_k^{(l)}}(\boldsymbol{x})$ with $\alpha_k^{(l)} \in \{1\} \cup (1 \pm 0.2)\hat{f}_*(S_k^{(l)})$ for every $k \in I_l$. We call $\mathcal{F}_l := \left\{S_k^{(l)}\right\}_{k \in I_l}$ the features contained in $\boldsymbol{x}^{(l)}$. Moreover, for any $S \in \mathcal{F}_l$ with $|S| \geq 2$, there is exactly one $k \in I_l$ with $S_k^{(l)} = S$.

(c) $\boldsymbol{x}^{(l)}$ contains all nonempty features in $\mathcal{K}_l$, i.e., $\mathcal{F}_l \supset \mathcal{K}_l \setminus \{\varnothing\}$.

The base case follows directly from $\boldsymbol{x}^{(1)} := \boldsymbol{x} \circ \boldsymbol{0}$. For the induction step, first, note that if $H \in \binom{[d+m]}{\leq D}$ contains an index that is not in I_l, then $z_H^{(l)}$ is identically zero. As a consequence, $w_H^{(l)}$ will stay at its initial value 0. Hence, it suffices to consider those entries with $H \subset I_l$. For notational simplicity, we define

$$\mathcal{H}_l := \{H \subset I_l \,:\, |H| \leq D\}. \tag{9}$$

This is the collection of indices of the nonzero entries of $\boldsymbol{z}^{(l)}$. By Assumption 3.1, for every $H \in \mathcal{H}_l$, we have

$$z_H^{(l)} = \textstyle\prod_{i \in H} \alpha_i^{(l)} \chi_{S_i^{(l)}}(\boldsymbol{x}) = \left(\prod_{i \in H} \alpha_i^{(l)}\right) \chi_{\bigoplus_{i \in H} S_i^{(l)}}(\boldsymbol{x}) =: \alpha_H^{(l)} \chi_{S_H^{(l)}}(\boldsymbol{x}), \tag{10}$$

where $\alpha_H^{(l)} := \prod_{i \in H} \alpha_i^{(l)} \neq 0$ and $S_H^{(l)} := \bigoplus_{i \in H} S_i^{(l)}$. We write $\mathcal{F}_{l+1/2} := \left\{S_H^{(l)}\right\}_{H \in \mathcal{H}_l}$ and call these subsets the features contained in $\boldsymbol{z}^{(l)}$. By the construction of $(\mathcal{K}_l)_l$ in (1), $\mathcal{F}_{l+1/2} \supset \mathcal{K}_{l+1}$.

It remains to select the features that are actually in $\mathcal{K}_{l+1}$ and remove duplicate representations. Deduplication is important, since duplicate representations will create more duplicate representations in later layers and quickly saturate the model capacity. We show that both feature selection and deduplication can be implemented simultaneously using one-pass coordinate descent (Algorithm 1).

The complete formal proof can be found in Appendix C. To see this intuitively, suppose that $S_H^{(l)} = S_{H'}^{(l)} = S$ for some $H \neq H' \in \mathcal{H}_l$. The only terms involving H and H' in $\left\langle \boldsymbol{w}^{(l)}, \boldsymbol{z}^{(l)} \right\rangle$ are

$$w_H^{(l)} z_H^{(l)} + w_{H'}^{(l)} z_{H'}^{(l)} = \left(w_H^{(l)} \alpha_H^{(l)} + w_{H'}^{(l)} \alpha_{H'}^{(l)}\right) \chi_S(\boldsymbol{x}),$$

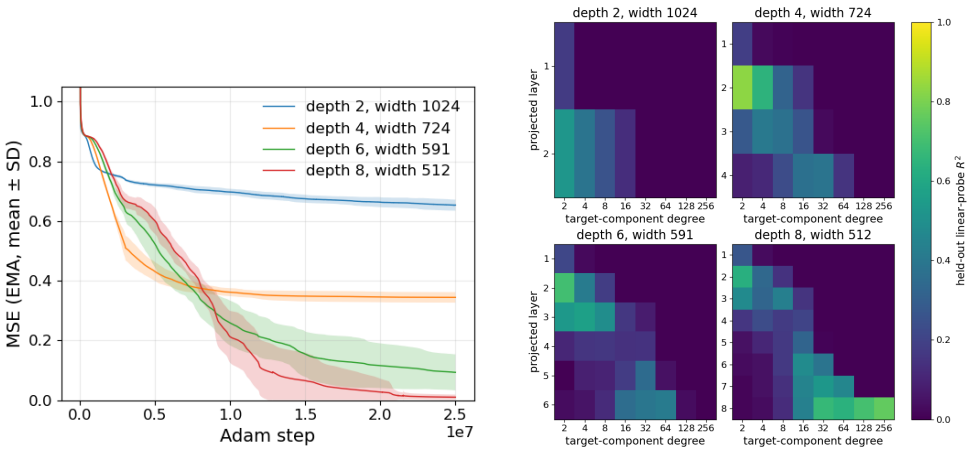
where the identity comes from (10). In particular, only $w_H^{(l)} \alpha_H^{(l)} + w_{H'}^{(l)} \alpha_{H'}^{(l)}$ matters. If we encounter H first (so $w_{H'}^{(l)}$ is still 0) and fit its coefficient exactly so that $w_H^{(l)} \alpha_H^{(l)} \chi_S(\boldsymbol{x}) = \hat{f}_*(S) \chi_S(\boldsymbol{x})$, then, when we reach H', the residual will be 0 and, therefore, $w_{H'}^{(l)}$ will stay at its initial value 0.

4 The Depth-Separation Theorem

In this section, we combine the results from the previous two sections and prove the separation theorem between constant- and logarithmic-depth networks. To show our theoretical claims can potentially be extended to more practical settings, we complement them with experiments on ReLU networks (cf. Figure 1), trained end-to-end using Adam (Kingma & Ba, 2017). Experiment details can be found in Appendix F.

To this end, we first formally define the class of deep quadratic functions.

[2] We do not need to track $\varnothing$ (which corresponds to the constant term in the Fourier decomposition of f_*), since the $\varnothing$-th entry in (4) always gives the constant term.



(a) Mean-squared error (MSE)

(b) Layer degree probing

Figure 1: Experiments on learning deep quadratic functions using ReLU networks. Figure 1a reports the (exponential moving average of) MSE. Figure 1b reports how well intermediate representations can be used to fit each level of the target deep quadratic function. One can observe that (a) trained deep ReLU networks can consistently reach near zero MSE, while shallow ones cannot, and (b) early layers tend to learn the low-degree terms and later layers tend to learn the high-degree ones.

Definition 4.1 (Deep quadratic function)**.** *Suppose $d = 2^{L_*}$ for some $L_* \in \mathbb{N}_+$ and let $d_l := 2^{-l}d$. We say a function $f_* : \{\pm 1\}^d \to \mathbb{R}$ is a deep quadratic function if it can be defined as*

$$f_*(\boldsymbol{x}) = L_*^{-1/2} \sum_{l=1}^{L_*} 2^{-(L_*-l)/2} \left\langle \mathbf{1}, \boldsymbol{x}^{(l)} \right\rangle, \quad \boldsymbol{x}^{(l)} = \left(x^{(l-1)}_{\sigma_l(2k-1)} x^{(l-1)}_{\sigma_l(2k)} \right)_{k \in [d_l]} \in \mathbb{R}^{d_l}, \quad \forall l \in [L_*],$$

where $\boldsymbol{x}^{(0)} = \boldsymbol{x} \in \mathbb{R}^{d_0}$ and each σ_l is an arbitrary permutation of $[d_{l-1}]$.

Fact 4.2. *Suppose that $d = 2^{L_*}$. Every deep quadratic function $f_* : \{\pm 1\}^d \to \mathbb{R}$ is a hierarchical staircase function with depth $L_* + 1$, local degree 2, width $d - 1$, and signal strength $\sqrt{2/(L_* d)}$. In addition, $\|f_*\|_{L^2} = 1$ and $\sum_{|S| \geq \sqrt{d}} \hat{f}_*^2(S) \geq 1/3$.*

By combining Theorem 2.1, Theorem 3.1, Fact 4.2, and Corollary D.2, we obtain the following theorem.

Theorem 4.1 (Depth separation)**.** *Suppose that $d = 2^{L_*}$ and that d is sufficiently large.*

(a) For any deep quadratic function $f_* : \{\pm 1\}^d \to \mathbb{R}$ and any constant-depth network satisfying Assumption 2.1, we have $\|f_L - f_*\|_{L^2(\mathrm{Unif}(\{\pm 1\}^d))} \geq 1/4$.

(b) Let $\varepsilon_*, \delta_{\mathbb{P}} \in (0, o(1))$ denote the target accuracy and failure probability, respectively. Consider the depth-(L_*+1) learner model in (5), with the model and algorithm parameters chosen according to Theorem 3.1 and Fact 4.2. Then, for any target deep quadratic function f_*, Algorithm 1 will output a network f satisfying $\|f_* - f\|_{L^2} \leq \varepsilon_*$ with probability at least $1 - \delta_{\mathbb{P}}$.

Moreover, we can efficiently compile f into a depth-$(L_* + 3)$ network satisfying Assumption 2.1(b)-(d) and $C_W, C_\phi = O(1)$.

5 Conclusion

In this work, we establish the first rigorous algorithmic separation between logarithmic-depth and constant-depth neural networks. We show that a depth-L network trained by layerwise coordinate descent efficiently learns depth-L hierarchical staircase functions by adaptively composing previously discovered Fourier features. In contrast, constant-depth networks satisfying our regularity and norm assumptions admit low-degree polynomial approximations and therefore cannot approximate deep quadratic functions, which place constant Fourier mass at high degrees. These results identify adaptive hierarchical feature construction as a mechanism through which depth enables efficient learning. An important direction is to extend this separation to end-to-end gradient-based training and more standard architectures while relaxing the assumptions required by the lower bound.

AI USE STATEMENT

In this work, we used generative AI tools for providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, and implementing methods. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, assisting in the writing of proofs, assisting with translation, or interpreting results. The following are not applicable to this work: generating synthetic data sets, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis. Additionally, we used generative AI tools for creating or editing software code, suggesting experimental parameters, sourcing/searching for information, editing a research paper to improve readability. We have reviewed all AI-assisted work. In particular, we verified and re-derived all LLM-generated arguments independently. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Emmanuel Abbe, Enric Boix-Adsera, Matthew S Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 26989–27002. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e2db7186375992e729165726762cb4c1-Paper.pdf.

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 *Proceedings of Thirty Fifth Conference on Learning Theory*, pp. 4782–4887. PMLR, June 2022. URL https://proceedings.mlr.press/v178/abbe22a.html.

Emmanuel Abbe, Enric Boix Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *Proceedings of Thirty Sixth Conference on Learning Theory*, pp. 2552–2623. PMLR, July 2023. URL https://proceedings.mlr.press/v195/abbe23a.html.

Zeyuan Allen-Zhu and Yuanzhi Li. What can ResNet learn efficiently, going beyond kernels? In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 809, pp. 9017–9028. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Zeyuan Allen-Zhu and Yuanzhi Li. Backward Feature Correction: How Deep Learning Performs Deep (Hierarchical) Learning. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 4598–4598. PMLR, July 2023. URL https://proceedings.mlr.press/v195/allen-zhu23a.html.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: a nonasymptotic theory of independence*. Oxford University Press, Oxford, 1st ed edition, 2013. ISBN 978-0-19-953525-5. OCLC: ocn818449985.

Joan Bruna. The Multiscale Single-Index Model: A Stylized Model for Hierarchical Feature Learning, July 2026. URL http://arxiv.org/abs/2607.03347. arXiv:2607.03347 [cs.LG].

Yatin Dandi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The Computational Advantage of Depth in Learning High-Dimensional Hierarchical Targets. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 160705–160764. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/eb3878c1dcbfff9ee95d5d033e5f5942-Paper-Conference.pdf.

Yatin Dandi, Matteo Vilucchio, Luca Arnaboldi, Hugo Tabanelli, and Florent Krzakala. Deep Learning as Neural Low-Degree Filtering: A Spectral Theory of Hierarchical Feature Learning, September 2026. URL http://arxiv.org/abs/2605.13612. arXiv:2605.13612 [cs.LG].

Amit Daniely. Depth Separation for Neural Networks. In Satyen Kale and Ohad Shamir (eds.), *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 690–696. PMLR, July 2017. URL https://proceedings.mlr.press/v65/daniely17a.html.

Amit Daniely. Deep Networks Learn Deep Hierarchical Models, January 2026. URL http://arxiv.org/abs/2601.00455. arXiv:2601.00455 [cs].

Ronen Eldan and Ohad Shamir. The Power of Depth for Feedforward Neural Networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir (eds.), *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 907–940, Columbia University, New York, New York, USA, June 2016. PMLR. URL http://proceedings.mlr.press/v49/eldan16.html.

Hengyu Fu, Zihao Wang, Eshaan Nichani, and Jason Lee. Learning Hierarchical Polynomials of Multiple Nonlinear Features. *International Conference on Learning Representations*, 2025:36166–36220, May 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/599221d7ebf6b3403190f38a3f282a1c-Abstract-Conference.html.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, November 1998. ISSN 0004-5411. doi: 10.1145/293347.293351. URL https://dl.acm.org/doi/10.1145/293347.293351.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs.LG].

Eshaan Nichani, Alex Damian, and Jason D Lee. Provable Guarantees for Nonlinear Feature Learning in Three-Layer Neural Networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10828–10875. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/236b6a814a1d2c0ff504ca7bf380f7ff-Paper-Conference.pdf.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 1 edition, June 2014. ISBN 978-1-107-03832-5 978-1-139-81478-2 978-1-107-47154-2. doi: 10.1017/CBO9781139814782. URL https://www.cambridge.org/core/product/identifier/9781139814782/type/book.

Yunwei Ren, Mo Zhou, and Rong Ge. Depth Separation with Multilayer Mean-Field Networks. In *The Eleventh International Conference on Learning Representations*, February 2023. URL https://openreview.net/forum?id=uzFQpkEzOo.

Yunwei Ren, Yatin Dandi, Florent Krzakala, and Jason D. Lee. Provable Learning of Random Hierarchy Models and Hierarchical Shallow-to-Deep Chaining. In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 5556–5597. PMLR, July 2026. URL https://proceedings.mlr.press/v336/ren26a.html.

Theodore J. Rivlin. *Chebyshev polynomials: from approximation theory to algebra and number theory*. Pure and applied mathematics. J. Wiley & sons, New York, 2nd ed edition, 1990. ISBN 978-0-471-62896-5.

Itay Safran and Jason Lee. Optimization-Based Separations for Neural Networks. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 3–64. PMLR, July 2022. URL https://proceedings.mlr.press/v178/safran22a.html.

Itay Safran and Ohad Shamir. Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2979–2987. PMLR, August 2017. URL https://proceedings.mlr.press/v70/safran17a.html.

Itay Safran, Daniel Reichman, and Paul Valiant. Depth Separations in Neural Networks: Separating the Dimension from the Accuracy. In *Proceedings of Thirty Eighth Conference on Learning Theory*, pp. 5108–5142. PMLR, July 2025. URL https://proceedings.mlr.press/v291/safran25a.html.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of Gradient-based Deep Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3067–3075, Sydney, NSW, Australia, August 2017. JMLR.org. URL https://dl.acm.org/doi/10.5555/3305890.3305998.

Hugo Tabanelli, Yatin Dandi, Luca Pesce, and Florent Krzakala. Deep Learning of Compositional Targets with Hierarchical Spectral Methods, February 2026. URL http://arxiv.org/abs/2602.10867. arXiv:2602.10867 [stat].

Matus Telgarsky. Benefits of depth in neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir (eds.), *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 1517–1539, Columbia University, New York, New York, USA, June 2016. PMLR. URL https://proceedings.mlr.press/v49/telgarsky16.html.

Lloyd N. Trefethen. *Approximation theory and approximation practice*. Number 164 in Other titles in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, extended edition edition, 2019. ISBN 978-1-61197-593-2 978-1-61197-594-9. doi: 10.1137/1.9781611975949.

Luca Venturi, Samy Jelassi, Tristan Ozuch, and Joan Bruna. Depth separation beyond radial functions. *Journal of Machine Learning Research*, 23(122):1–56, 2022. URL http://jmlr.org/papers/v23/21-1109.html.

Zihao Wang and Lei Wu. Theoretical analysis of inductive biases in deep convolutional networks, 2024. URL https://arxiv.org/abs/2305.08404.

Zihao Wang, Eshaan Nichani, and Jason D. Lee. Learning Hierarchical Polynomials with Three-Layer Neural Networks. In *The Twelfth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=QgwAYFrh9t&referrer=%5Bthe%20profile%20of%20Zihao%20Wang%5D(%2Fprofile%3Fid%3D~Zihao_Wang25).

Greg Yang and Edward J. Hu. Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11727–11737. PMLR, July 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

Greg Yang, James B. Simon, and Jeremy Bernstein. A Spectral Condition for Feature Learning, May 2024. URL http://arxiv.org/abs/2310.17813. arXiv:2310.17813 [cs.LG].

CONTENTS

1 Introduction **1**
1.1 Our contributions . . . 2
1.2 Related work . . . 3
1.3 Notation and conventions . . . 3
1.4 Outline . . . 3

2 Lower bound against constant-depth networks **4**
2.1 Setup and main result . . . 4
2.2 Proof sketch . . . 5

3 Upper bound for logarithmic-depth networks **6**
3.1 Setup and main result . . . 6
3.2 Proof sketch . . . 7

4 The depth-separation theorem **8**

5 Conclusion **9**

A Chebyshev polynomials and approximation **13**

B Lower bound proofs **16**
B.1 Tail bounds for the intermediate representations . . . 16
B.2 Approximating the network . . . 19
B.3 Proof of the main lower bound theorem . . . 26

C Upper bound proofs **26**

D Compiling the learner network **31**

E Further discussion and examples of hierarchical staircase functions **33**
E.1 Staircase-type functions . . . 33
E.2 Globally read-once decision trees . . . 34
E.3 Addressing functions . . . 35

F Experimental details **35**

A CHEBYSHEV POLYNOMIALS AND APPROXIMATION

Definition A.1 (Chebyshev polynomials)**.** *The* **Chebyshev polynomials (of the first kind)** $(T_n)_{n\in\mathbb{N}_0}$ *are the polynomials with real coefficients defined by*

$$T_n(x) = \begin{cases} \cos(n \arccos x), & |x| \le 1, \\ \cosh(n \operatorname{arccosh} x), & x \ge 1 \\ (-1)^n \cosh(n \operatorname{arccosh}(-x)), & x \le -1. \end{cases}$$

Remark. Equivalently, Chebyshev polynomials can be defined via the recurrence relation $T_0(x) = 1$, $T_1(x) = x$, $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, which makes it obvious that T_n is a degree-n polynomial. ♣

Fact A.2. *For any $n \in \mathbb{N}_0$ and $x \in \mathbb{R}$, we have $|T_n(x)| \le (1 + 2|x|)^n$.*

Proof. For $x \in [-1, 1]$, it is clear that $|T_n(x)| \le 1$. When $|x| \ge 1$, we have

$$\begin{aligned}|T_n(x)| &= \cosh(n \operatorname{arccosh}|x|) \le \exp(n \operatorname{arccosh}|x|) \\ &= \exp\left(n \log\left(|x| + \sqrt{x^2 - 1}\right)\right) \le \exp(n \log(2|x|)) = (2|x|)^n.\end{aligned}$$

□

Lemma A.3 (Equation (2.37) of Rivlin (1990))**.** *For any degree-n polynomial $f : \mathbb{R} \to \mathbb{R}$ with $\|f\|_{L^\infty([-1,1])} \le 1$, we have $|f(x)| \le |T_n(x)|$ for all $|x| \ge 1$. Combining this with Fact A.2, we obtain $|f(x)| \le (1 + 2|x|)^n$.*

Definition A.4 (Bernstein ellipse)**.** *Let $\rho > 1$. The* **Bernstein ellipse** *with parameter ρ is defined as*

$$E_\rho := \left\{(w + w^{-1})/2 : w \in \mathbb{C}, |w| = \rho\right\} \subset \mathbb{C}.$$

In other words, E_ρ is the ellipse in $\mathbb{C}$ with foci ± 1 and semi-major and semi-minor axes $(\rho \pm \rho^{-1})/2$. We use E_ρ^o to denote the open region enclosed by E_ρ.

Lemma A.5 (Theorem 8.2 of Trefethen (2019))**.** *Let $\rho > 1$. Suppose that $f : \mathbb{R} \to \mathbb{R}$ is real analytic, can be analytically continued to E_ρ, and is bounded by M in E_ρ. Then, its Chebyshev coefficients $(a_p)_p$ satisfy*

$$|a_p| \le 2M\rho^{-p}, \quad \forall p \in \mathbb{N}.$$

In particular, this implies that its degree-p Chebyshev approximation $\tilde{f}$ satisfies

$$\left\|f - \tilde{f}\right\|_{L^\infty([-1,1])} \le \frac{2M}{\rho - 1}\rho^{-p}.$$

Lemma A.6. *Consider $\phi : \mathbb{R} \to \mathbb{R}$. Suppose that for some $c_0 > 0$, ϕ can be analytically continued to the strip $\{z \in \mathbb{C} : |\operatorname{Im} z| < c_0\}$, and ϕ has at most polynomial growth:*

$$|\phi(a + ib)| \le K(1 + |a|)^m, \quad a, b \in \mathbb{R}, |b| \le c_0/2,$$

for some $K, m \ge 0$.

Fix $x_0 \in \mathbb{R}$ and $R > 0$ and define $\psi(u) := \phi(x_0 + Ru)$. Let $\psi(u) = \sum_{n=0}^\infty a_n T_n(u)$ be the Chebyshev expansion of ψ on $[-1, 1]$ and define $\tilde{\psi}_p := \sum_{n=0}^p a_n T_n$. We have $|a_n| \le 2M_R\rho_R^{-n}$ for every $n \in \mathbb{N}$, and

$$\left\|\psi - \tilde{\psi}_p\right\|_{L^\infty([-1,1])} \le \frac{2M_R}{\rho_R - 1}\rho_R^{-p},$$

where

$$\rho_R := 1 + \frac{c_0}{2R}, \quad M_R := K\left(1 + |x_0| + R + c_0/2\right)^m.$$

In particular, for any target accuracy $\varepsilon > 0$, if

$$R \ge \frac{c_0}{2}, \quad p \ge \frac{4R}{c_0}\log\left(\frac{4RM_R}{c_0\varepsilon}\right),$$

the degree-p polynomial $\tilde{\phi}(x) := \tilde{\psi}_p((x - x_0)/R)$ satisfies

$$\left\|\phi - \tilde{\phi}\right\|_{L^\infty([x_0 - R, x_0 + R])} = \left\|\psi - \tilde{\psi}_p\right\|_{L^\infty([-1,1])} \le \varepsilon.$$

Proof. First, we show that $z \mapsto \psi(z) = \phi(x_0 + Rz)$ is analytic in E_{ρ_R}. For any $z \in E_{\rho_R}^o$, since the semi-minor axis of E_{ρ_R} is $(\rho_R - \rho_R^{-1})/2$, we have

$$|\operatorname{Im}(x_0 + Rz)| = R|\operatorname{Im} z| \le R(\rho_R - \rho_R^{-1})/2 \le c_0/2.$$

Since ϕ is analytic in the strip of half-width c_0, this implies ψ is analytic in E_{ρ_R}. To apply Lemma A.5, we also need to bound the maximum magnitude of ψ in E_{ρ_R}. Since the semi-major axis of E_{ρ_R} is $(\rho_R + \rho_R^{-1})/2 \le \rho_R = 1 + c_0/(2R)$, for any $z \in E^o_{\rho_R}$, we have

$$|\mathrm{Re}(x_0 + Rz)| \le |x_0| + R|\mathrm{Re}\, z| \le |x_0| + R(1 + c_0/(2R)) = |x_0| + R + c_0/2.$$

Hence, by the polynomial growth of ϕ, we have

$$|\psi(z)| = |\phi(x_0 + Rz)| \le K(1 + |x_0| + R + c_0/2)^m =: M_R, \quad \forall z \in E^o_{\rho_R}.$$

Thus, by Lemma A.5, we have

$$|a_n| \le 2M_R\rho_R^{-n}, \quad \left\|\psi - \tilde{\psi}_p\right\|_{L^\infty([-1,1])} \le \frac{2M_R}{\rho_R - 1}\rho_R^{-p}.$$

For the second part of the lemma, note that when $R \ge c_0/2$, we have $c_0/(2R) \le 1$ and therefore, $1 + \rho_R \le 3$ and $\log \rho_R = \log(1 + c_0/(2R)) \ge c_0/(4R)$. As a result,

$$\frac{2M_R}{\rho_R - 1}\rho_R^{-p} \le \varepsilon \quad \Leftarrow \quad p \log \rho_R \ge \log\left(\frac{4RM_R}{c_0\varepsilon}\right) \quad \Leftarrow \quad p \ge \frac{4R}{c_0}\log\left(\frac{4RM_R}{c_0\varepsilon}\right).$$

□

Lemma A.7 (Common activation functions)**.** *The following activation functions satisfy the assumptions of Lemma A.6 with $c_0 = \pi/2$, $K \le 4$ and $m \le 1$:*

$$\tanh x, \quad \mathrm{Sigmoid}(x) := \frac{1}{1 + e^{-x}}, \quad \mathrm{Softplus}(x) := \log(1 + e^x), \quad \mathrm{GELU}(x) := x\Phi(x),$$

where Φ is the standard Gaussian CDF.

Proof. In this proof, we call $\{z \in \mathbb{C} : |\mathrm{Im}\, z| < c_0\}$ the c_0-strip.

The function $\tanh z$ is meromorphic with poles at $i\pi(k + 1/2)$, $k \in \mathbb{Z}$, and is therefore analytic in the $\pi/2$-strip. Moreover, for $|b| \le \pi/4$,

$$|\tanh(a + ib)|^2 = \frac{\sin^2(2b) + \sinh^2(2a)}{(\cos(2b) + \cosh(2a))^2} \le \frac{1 + \sinh^2(2a)}{\cosh^2(2a)} = \frac{1}{\cosh^2(2a)} + \tanh^2(2a) \le 2,$$

so the strip-growth condition holds with $K = 2$ and $m = 0$.

Sigmoid has possible poles only where $1 + e^{-z} = 0$, namely at $(2k + 1)i\pi$, $k \in \mathbb{Z}$. Thus, it is analytic in the π-strip. If $|b| \le \pi/2$, then $\mathrm{Re}(e^{-a-ib}) = e^{-a}\cos b \ge 0$, and hence $|1 + e^{-a-ib}| \ge 1$. Therefore, $|\mathrm{Sigmoid}(a + ib)| \le 1$. Hence, sigmoid satisfies the strip-growth condition with $K = 1$ and $m = 0$.

Now consider softplus. Again, since the only zeros of $1 + e^z$ are at $(2k+1)i\pi$, $k \in \mathbb{Z}$, $\log(1 + e^z)$ is analytic in the π-strip. Hence, we can take the branch of $\log(1+e^z)$ that matches the real $\log(1+e^x)$ on the real line. Now, consider $z = a + ib$ with $a, b \in \mathbb{R}$, $|b| \le \pi/2$. As in the preceding argument, $\mathrm{Re}(1 + e^z) \ge 1$. Therefore, we have

$$|\mathrm{Re}\log(1 + e^z)| = \log|1 + e^z| \le \log(1 + e^a) \le \log 2 + |a|.$$

For the imaginary part, we have

$$|\mathrm{Im}\log(1 + e^z)| = |\mathrm{Arg}\,(1 + e^z)| \le |b| \le \pi/2.$$

As a result, on the $\pi/2$-strip, we have

$$|\log(1 + e^z)| \le \sqrt{(\log 2 + |a|)^2 + (\pi/2)^2} \le \pi/2 + \log 2 + |a|.$$

Hence, softplus satisfies the strip-growth condition with $K = 4$ and $m = 1$.

Finally, consider the GELU activation $\mathrm{GELU}(x) = x\Phi(x)$. Since $w \mapsto e^{-w^2/2}$ is entire, so is $\Phi(z) = 1/2 + (2\pi)^{-1/2}\int_0^z e^{-w^2/2}\,\mathrm{d}w$. Hence, GELU can be analytically continued to any strip.

For simplicity, take the π-strip. For $z = a + ib$ with $a, b \in \mathbb{R}, |b| \le \pi/4$, to bound $\Phi(z)$, we first integrate $e^{-w^2/2}$ from 0 to a and then from a to $a + ib$. This gives

$$|\Phi(z)| \le \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \int_0^{|a|} e^{-t^2/2}\,\mathrm{d}t + \frac{1}{\sqrt{2\pi}} \int_0^{|b|} \left| e^{-(a+is)^2/2} \right| \mathrm{d}s$$
$$\le 1 + \frac{1}{\sqrt{2\pi}} \int_0^{\pi/2} e^{s^2/2}\,\mathrm{d}s \le 4.$$

As a result, we have

$$|\mathrm{GELU}(z)| \le 4|z| \le 4\,(1 + |a|)\,.$$

Thus, it satisfies the strip-growth condition with $K = 4$ and $m = 1$. □

B LOWER BOUND PROOFS

In this section, we prove Theorem 2.1. As discussed in the proof sketch, it suffices to show that any network satisfying Assumption 2.1 can be approximated by a polynomial of degree at most $d^{1/2}$. Formally, we will prove the following lemma and then derive Theorem 2.1 as a corollary of it.

Lemma 2.2. *Suppose that Assumption 2.1 holds. Then, for any $\varepsilon_* \ge d^{-O(1)}$, there exists a polynomial $\tilde{f}_L : \{\pm 1\}^d \to \mathbb{R}$ such that $\left\| f_L - \tilde{f}_L \right\|_{L^2} \le \varepsilon_*$ and*

$$\deg \tilde{f}_L \lesssim \left((C_\phi C_W^2)^{2^{L+3}} (\log d)^{2L+2} \right)^{L(2^L+1)} \ll d^{1/2}.$$

The rest of this section is organized as follows. First, we show that the true intermediate representations $\boldsymbol{x}^{(l)}$ are subgaussian in Appendix B.1. Then, in Appendix B.2, we prove Lemma 2.2 by approximating each layer of the learner network with a polynomial. Finally, in Appendix B.3, we prove Theorem 2.1.

B.1 TAIL BOUNDS FOR THE INTERMEDIATE REPRESENTATIONS

In this subsection, we show that the $\boldsymbol{x}^{(l)}$ are subgaussian. The Gaussian Lipschitz-concentration argument does not apply here because vectors such as $\boldsymbol{W}^{(1)}\boldsymbol{x}^{(1)}$ are subgaussian but need not be Gaussian. Instead, we use the following adaptive version of McDiarmid's inequality.

Lemma B.1 (Theorem 6.5 of Boucheron et al. (2013))**.** *Let $\mathcal{X}$ be the domain of each input coordinate. Suppose that $f : \mathcal{X}^d \to \mathbb{R}$ satisfies the following $\boldsymbol{x}$-dependent bounded-difference property: there exist $c_1, \dots, c_d : \mathcal{X}^{d-1} \to [0, \infty)$ such that for any $\boldsymbol{x} \in \mathcal{X}^d$,*

$$\sup_{y, y' \in \mathcal{X}} |f(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_d) - f(x_1, \dots, x_{i-1}, y', x_{i+1}, \dots, x_d)| \le c_i(\boldsymbol{x}_{-i}), \quad \forall i \in [d],$$

where $\boldsymbol{x}_{-i}$ denotes the vector obtained by removing the i-th coordinate of $\boldsymbol{x}$. If $(1/4)\sum_{i=1}^d c_i^2(\boldsymbol{x}_{-i}) \le v$ for some $v > 0$, then we have

$$\mathbb{P}_{\boldsymbol{x}} \left[|f(\boldsymbol{x}) - \mathbb{E}\, f(\boldsymbol{x})| \ge t \right] \le 2e^{-t^2/(2v)}, \quad \forall t > 0,$$

where $\boldsymbol{x} \in \mathcal{X}^d$ is any random vector with independent entries.

Remark. When the domain is $\{\pm 1\}^d$, since $x_i \in \{\pm 1\}$, we can always choose

$$c_i(\boldsymbol{x}_{-i}) := |f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_d) - f(x_1, \dots, x_{i-1}, -1, x_{i+1}, \dots, x_d)| = |f(\boldsymbol{x}) - f(\boldsymbol{x}_{\oplus i})|,$$

where $\boldsymbol{x} \in \{\pm 1\}^d$ is any vector consistent with $\boldsymbol{x}_{-i}$ and $\boldsymbol{x}_{\oplus i}$ denotes the vector obtained by flipping the i-th coordinate of $\boldsymbol{x}$. Therefore, it suffices to upper bound

$$\sup_{\boldsymbol{x} \in \{\pm 1\}^d} \frac{1}{4} \sum_{i=1}^d \left(f(\boldsymbol{x}) - f(\boldsymbol{x}_{\oplus i}) \right)^2 =: \sup_{\boldsymbol{x} \in \{\pm 1\}^d} \frac{1}{4} V_f(\boldsymbol{x}).$$

♣

To illustrate how this lemma will be used, consider $f_l(\boldsymbol{x}) := \left\langle \boldsymbol{a}^{(l)}, \boldsymbol{x}^{(l+1)}(\boldsymbol{x}) \right\rangle$. Write

$$V_{f_l}(\boldsymbol{x}) = \sum_{k=1}^{d} \left\langle \boldsymbol{a}^{(l)}, \boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) \right\rangle^2 = \left\| (\boldsymbol{a}^{(l)})^\top \left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) \right]_{k\in[d]} \right\|^2$$
$$\leq \left\| \boldsymbol{a}^{(l)} \right\|_2^2 \left\| \left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) \right]_{k\in[d]} \right\|_{\mathrm{OP}}^2 .$$

For the second factor, recall that $\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) = \boldsymbol{\phi}_l(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}))$. Hence, by Taylor expanding $\phi_{l,k}$ at $\left\langle \boldsymbol{w}_k^{(l)}, \boldsymbol{x}^{(l)} \right\rangle$, we get

$$\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) = \boldsymbol{\phi}_l \left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}) \right) - \boldsymbol{\phi}_l \left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}) + \boldsymbol{W}^{(l)} \left(\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x}) \right) \right)$$
$$\approx \operatorname{diag} \left(\boldsymbol{\phi}_l' \left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}) \right) \right) \boldsymbol{W}^{(l)} \left(\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x}) \right),$$

where $\boldsymbol{\phi}_l' = (\phi_{l,k}')_k$. As a result, we have

$$\left\| \left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) \right]_{k\in[d]} \right\|_{\mathrm{OP}} \lesssim \left\| \operatorname{diag} \left(\boldsymbol{\phi}_l' \left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}) \right) \right) \boldsymbol{W}^{(l)} \left[\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x}) \right]_{k\in[d]} \right\|_{\mathrm{OP}}$$
$$\leq C_\phi C_W \left\| \left[\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x}) \right]_{k\in[d]} \right\|_{\mathrm{OP}} .$$

Note that the last factor is exactly the left-hand side with $l+1$ replaced by l. Thus, we can iterate the above argument to obtain an upper bound on the operator norm of $\left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) \right]_{k\in[d]}$. Combining this with Lemma B.1, we conclude that $f^{(l)}(\boldsymbol{x})$ is subgaussian for every $l \in [L]$.

To formalize the above argument, we first prove the following deterministic lemma.

Lemma B.2. *Let $\boldsymbol{y} \in \mathbb{R}^{d''}$ and $\boldsymbol{M} \in \mathbb{R}^{d''\times d'}$ be arbitrary, and let $\phi_1, \ldots, \phi_{d''} : \mathbb{R} \to \mathbb{R}$ be C^2 functions satisfying $\|\phi_k'\|_{L^\infty} \vee \|\phi_k''\|_{L^\infty} \leq C_\phi$ for every $k \in [d'']$. Write $\boldsymbol{\phi}(\boldsymbol{y}) := (\phi_i(y_i))_{i\in[d'']}$ and define*

$$\mathcal{A}(\boldsymbol{y}; \boldsymbol{M}) := [\boldsymbol{\phi}(\boldsymbol{y} + \boldsymbol{M}_{:,1}) - \boldsymbol{\phi}(\boldsymbol{y}) \quad \cdots \quad \boldsymbol{\phi}(\boldsymbol{y} + \boldsymbol{M}_{:,d'}) - \boldsymbol{\phi}(\boldsymbol{y})] \in \mathbb{R}^{d''\times d'}.$$

Then, we have

$$\|\mathcal{A}(\boldsymbol{y}; \boldsymbol{M})\|_{\mathrm{OP}} \leq C_\phi \|\boldsymbol{M}\|_{\mathrm{OP}} + \frac{C_\phi}{2} \|\boldsymbol{M}\|_{\mathrm{OP}}^2 .$$

Proof. Consider the (i, j) entry of $\mathcal{A}(\boldsymbol{y}; \boldsymbol{M})$. We have

$$\phi_i(y_i + M_{i,j}) - \phi_i(y_i) = \phi_i'(y_i)M_{i,j} + \frac{R_{i,j}}{2} M_{i,j}^2,$$

where $R_{i,j} \in \mathbb{R}$ satisfies $|R_{i,j}| \leq \|\phi_i''\|_{L^\infty} \leq C_\phi$. Therefore, we can write $\mathcal{A}(\boldsymbol{y}; \boldsymbol{M})$ as

$$\mathcal{A}(\boldsymbol{y}; \boldsymbol{M}) = \operatorname{diag}(\boldsymbol{\phi}'(\boldsymbol{y}))\boldsymbol{M} + \frac{1}{2}\boldsymbol{R} \odot \boldsymbol{M}^{\odot 2},$$

where $\odot$ denotes the Hadamard product and $\boldsymbol{M}^{\odot 2}$ denotes the entrywise square. Thus, we have

$$\|\mathcal{A}(\boldsymbol{y}; \boldsymbol{M})\|_{\mathrm{OP}} \leq \|\operatorname{diag}(\boldsymbol{\phi}'(\boldsymbol{y}))\|_{\mathrm{OP}} \|\boldsymbol{M}\|_{\mathrm{OP}} + \frac{1}{2} \left\|\boldsymbol{R} \odot \boldsymbol{M}^{\odot 2}\right\|_{\mathrm{OP}} \leq C_\phi \|\boldsymbol{M}\|_{\mathrm{OP}} + \frac{1}{2} \left\|\boldsymbol{R} \odot \boldsymbol{M}^{\odot 2}\right\|_{\mathrm{OP}} .$$

For the second term, note that since $\boldsymbol{M}^{\odot 2}$ is entrywise nonnegative, for any unit vector $\boldsymbol{u}, \boldsymbol{v}$, we have

$$\left|\boldsymbol{u}^\top \left(\boldsymbol{R} \odot \boldsymbol{M}^{\odot 2}\right) \boldsymbol{v}\right| = \left| \sum_{i,j} u_i v_j R_{ij} M_{ij}^2 \right| \leq C_\phi \sum_{i,j} |u_i||v_j| M_{ij}^2$$
$$= C_\phi |\boldsymbol{u}|^\top \boldsymbol{M}^{\odot 2} |\boldsymbol{v}| \leq C_\phi \left\|\boldsymbol{M}^{\odot 2}\right\|_{\mathrm{OP}} \leq C_\phi \|\boldsymbol{M}\|_{\mathrm{OP}}^2 .$$

In other words, $\left\|\boldsymbol{R} \odot \boldsymbol{M}^{\odot 2}\right\|_{\mathrm{OP}} \leq C_\phi \|\boldsymbol{M}\|_{\mathrm{OP}}^2$. □

Lemma B.3. *Suppose that Assumption 2.1 holds. For $l \in [L]$, define*

$$\mathcal{A}_l := \left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x})\right]_{k\in[d]} \in \mathbb{R}^{d_{l+1}\times d}, \quad \mathcal{A}_0 := [\boldsymbol{x}_{\oplus k} - \boldsymbol{x}]_{k\in[d]} \in \mathbb{R}^{d_1\times d}.$$

Then, we have $\|\mathcal{A}_0\|_{\mathrm{OP}} = 2$ and

$$\|\mathcal{A}_l\|_{\mathrm{OP}} \le C_\phi C_W \|\mathcal{A}_{l-1}\|_{\mathrm{OP}} + \frac{C_\phi C_W^2}{2}\|\mathcal{A}_{l-1}\|_{\mathrm{OP}}^2, \quad \forall l \in [L].$$

As a corollary, we have $\|\mathcal{A}_l\|_{\mathrm{OP}} \le (4C_\phi C_W^2)^{2^l}$ for all $l \in [L]$.

Proof. Recall that $\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) := \boldsymbol{\phi}_l(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}))$. Write

$$\begin{aligned}\mathcal{A}_l &= \left[\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k})\right) - \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x})\right)\right]_{k\in[d]} \\ &= \left[\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x}) + \boldsymbol{W}^{(l)}\left(\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x})\right)\right) - \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x})\right)\right]_{k\in[d]}.\end{aligned}$$

Apply Lemma B.2 with $\boldsymbol{y} = \boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}(\boldsymbol{x})$ and $\boldsymbol{M} = \boldsymbol{W}^{(l)}\left(\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x})\right)_k$, note that

$$\|\boldsymbol{M}\|_{\mathrm{OP}} \le \left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}} \left\|\left[\boldsymbol{x}^{(l)}(\boldsymbol{x}_{\oplus k}) - \boldsymbol{x}^{(l)}(\boldsymbol{x})\right]_k\right\|_{\mathrm{OP}} \le C_W \|\mathcal{A}_{l-1}\|_{\mathrm{OP}},$$

and we obtain

$$\begin{aligned}\|\mathcal{A}_l\|_{\mathrm{OP}} &\le C_\phi \left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}} \|\mathcal{A}_{l-1}\|_{\mathrm{OP}} + \frac{\|\phi_l''\|_{L^\infty}}{2}\left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}}^2 \|\mathcal{A}_{l-1}\|_{\mathrm{OP}}^2 \\ &\le C_\phi C_W \|\mathcal{A}_{l-1}\|_{\mathrm{OP}} + \frac{C_\phi C_W^2}{2}\|\mathcal{A}_{l-1}\|_{\mathrm{OP}}^2.\end{aligned}$$

In particular, when $\|\mathcal{A}_{l-1}\|_{\mathrm{OP}} \ge 1$, the above bound implies

$$\|\mathcal{A}_l\|_{\mathrm{OP}} \le 2C_\phi C_W^2 \|\mathcal{A}_{l-1}\|_{\mathrm{OP}}^2.$$

When $\|\mathcal{A}_{l-1}\|_{\mathrm{OP}} \le 1$, it suffices to replace $\|\mathcal{A}_{l-1}\|_{\mathrm{OP}}$ with $\|\mathcal{A}_{l-1}\|_{\mathrm{OP}} \vee 1$ in the above recurrence. Thus, we have $\|\mathcal{A}_l\|_{\mathrm{OP}} \le (4C_\phi C_W^2)^{2^l}$. □

Lemma B.4 (Subgaussianity of the intermediate representations)**.** *Suppose that Assumption 2.1 holds. Then, for any $l \in [L]$ and $\boldsymbol{a} \in \mathbb{S}^{d_{l+1}-1}$, we have, for every $t \ge 0$,*

$$\mathbb{P}\left[\left|\left\langle \boldsymbol{a}, \boldsymbol{x}^{(l+1)}(\boldsymbol{x})\right\rangle - \mathbb{E}\left\langle \boldsymbol{a}, \boldsymbol{x}^{(l+1)}(\boldsymbol{x})\right\rangle\right| \ge t\right] \le 2\exp\left(-\frac{t^2}{(4C_\phi C_W^2)^{2^{l+1}}}\right) \le 2\exp\left(-\frac{t^2}{2\sigma_X^2}\right),$$

where $\sigma_X = (4C_\phi C_W^2)^{2^L}$. In other words, $\boldsymbol{x}^{(l+1)} - \mathbb{E}\,\boldsymbol{x}^{(l+1)}$ has a σ_X^2-subgaussian tail.

Proof. By the argument following Lemma B.1, we have

$$V_{\langle \boldsymbol{a}, \boldsymbol{x}^{(l+1)}\rangle}(\boldsymbol{x}) \le \|\boldsymbol{a}\|^2 \left\|\left[\boldsymbol{x}^{(l+1)}(\boldsymbol{x}) - \boldsymbol{x}^{(l+1)}(\boldsymbol{x}_{\oplus k})\right]_{k\in[d]}\right\|_{\mathrm{OP}}^2 \le (4C_\phi C_W^2)^{2^{l+1}},$$

where the second inequality comes from Lemma B.3. To complete the proof, it suffices to invoke Lemma B.1. □

We close this subsection with the following estimate on $\mathbb{E}\,\boldsymbol{x}^{(l+1)}(\boldsymbol{x})$.

Lemma B.5. *Suppose that Assumption 2.1 holds. Then, for any $l \in [L]$, we have*

$$\left\|\mathbb{E}\,\boldsymbol{x}^{(l+1)}\right\| \le 2L\sigma_X(C_\phi C_W)^L\sqrt{d_{\max}} =: C_X\sqrt{d_{\max}},$$

where $C_X := 2L\sigma_X(C_\phi C_W)^L$.

Proof. Recall that $\boldsymbol{x}^{(l+1)} = \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}\right)$. First, by the Lipschitzness of $\boldsymbol{\phi}_l$, we have

$$\left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}\right) - \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\,\mathbb{E}[\boldsymbol{x}^{(l)}]\right)\right\| \le C_\phi C_W \left\|\boldsymbol{x}^{(l)} - \mathbb{E}[\boldsymbol{x}^{(l)}]\right\|.$$

Therefore,

$$\begin{aligned}\left\|\mathbb{E}\,\boldsymbol{x}^{(l+1)}\right\| &\le \left\|\mathbb{E}\,\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l)}\right) - \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\,\mathbb{E}[\boldsymbol{x}^{(l)}]\right)\right\| + \left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\,\mathbb{E}[\boldsymbol{x}^{(l)}]\right)\right\| \\ &\le C_\phi C_W\,\mathbb{E}\left\|\boldsymbol{x}^{(l)} - \mathbb{E}[\boldsymbol{x}^{(l)}]\right\| + \left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\,\mathbb{E}[\boldsymbol{x}^{(l)}]\right)\right\|.\end{aligned}$$

For the first term, by Lemma B.4, we have $C_\phi C_W\,\mathbb{E}\left\|\boldsymbol{x}^{(l)} - \mathbb{E}[\boldsymbol{x}^{(l)}]\right\| \le C_\phi C_W \sigma_X \sqrt{d_{\max}}$. For the second term, by Assumption 2.1, we have

$$\left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\,\mathbb{E}[\boldsymbol{x}^{(l)}]\right)\right\| \le \|\boldsymbol{\phi}_l(0)\| + C_\phi C_W \left\|\mathbb{E}[\boldsymbol{x}^{(l)}]\right\| \le C_\phi\sqrt{d_{\max}} + C_\phi C_W \left\|\mathbb{E}[\boldsymbol{x}^{(l)}]\right\|.$$

As a result, we have

$$\left\|\mathbb{E}\,\boldsymbol{x}^{(l+1)}\right\| \le 2C_\phi C_W \sigma_X \sqrt{d_{\max}} + C_\phi C_W \left\|\mathbb{E}[\boldsymbol{x}^{(l)}]\right\|, \quad \left\|\mathbb{E}\,\boldsymbol{x}^{(1)}\right\| = \|\mathbb{E}\,\boldsymbol{x}\| = 0.$$

Solving the above recurrence completes the proof. □

B.2 Approximating the network

Notation. We use $\|\cdot\|_{L^{p,q}}$ to denote the following norm. Let $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ be an arbitrary (integrable) vector-valued function. For any $p, q \in [1, \infty]$, we define

$$\|\boldsymbol{h}\|_{L^{p,q}} = \left\|\|\boldsymbol{h}(\boldsymbol{x})\|_p\right\|_{L^q_{\boldsymbol{x}}} = \left(\mathbb{E}\,\|\boldsymbol{h}(\boldsymbol{x})\|_p^q\right)^{1/q}.$$

That is, $\|\boldsymbol{h}\|_{L^{p,q}}$ is the L^q norm of the ℓ_p-norm of $\boldsymbol{h}$. When $p = 2$, we will simply write $\|\boldsymbol{h}\|_{L^q}$.

Recall from Section 2 that our approximating network $\tilde{f}_L$ is constructed by replacing each activation function $\phi_{l,k}$ with its degree-p_l Chebyshev approximation $\tilde{\phi}_{l,k}$ over the interval $I_k^{(l)} := [\bar{z}_k^{(l)} - R_l, \bar{z}_k^{(l)} + R_l]$, where $\bar{\boldsymbol{z}}^{(l)} := \boldsymbol{W}^{(l)}\,\mathbb{E}\,\boldsymbol{x}^{(l)}$ and p_l is chosen so that the L^∞-difference over $I_k^{(l)}$ is bounded by ε_ϕ. In this subsection, we show that we can choose $(R_l)_l$ and ε_ϕ so that $f_L \approx \tilde{f}_L$ and $p_l = d^{o(1)}$. We will use Lemma B.4 and Lemma B.5 extensively in this subsection. For convenience, we restate them here: Under Assumption 2.1, we have, for every $l \in [L]$,

$$\boldsymbol{x}^{(l+1)} - \mathbb{E}\,\boldsymbol{x}^{(l+1)} \text{ has a } \sigma_X^2\text{-subgaussian tail and } \left\|\mathbb{E}\,\boldsymbol{x}^{(l+1)}\right\| \le C_X\sqrt{d_{\max}}, \tag{11}$$

where $\sigma_X := (4C_\phi C_W^2)^{2^L}$ and $C_X := 2L\sigma_X(C_\phi C_W)^L$.

As discussed in the proof sketch, we will use a layerwise argument. Formally, let $\mathcal{G}_l$ denote the event that in the approximating network, all preactivations at layer l lie in their corresponding approximation intervals $I_k^{(l)}$. That is,

$$\mathcal{G}_l := \left\{\boldsymbol{x} \in \{\pm 1\}^d : \max_{k\in[d_{l+1}]} \left|\left\langle \boldsymbol{w}_k^{(l)}, \tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right\rangle - \bar{z}_k^{(l)}\right| \le R_l\right\}.$$

For notational simplicity, we also define $\mathcal{G}_{<l} = \mathcal{G}_1 \cap \cdots \cap \mathcal{G}_{l-1}$ and $\mathcal{G}_{\le l} = \mathcal{G}_{<l} \cap \mathcal{G}_l$. Then, we define $\mathcal{B}_l$ to be the event that the first failure happens at layer l. Namely,

$$\mathcal{B}_l := \mathcal{G}_{<l} \cap \mathcal{G}_l^c = \mathcal{G}_1 \cap \cdots \cap \mathcal{G}_{l-1} \cap \mathcal{G}_l^c.$$

Note that the input space can be decomposed as $\{\pm 1\}^d = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_l \cup \mathcal{G}_{\le l}$ for every $l \in [L]$. Now, we decompose the error according to this decomposition of input space.

Lemma B.6. *Suppose that Assumption 2.1 holds. Then, we have*

$$\left\|f_L - \tilde{f}_L\right\|_{L^2} \le L(C_\phi C_W)^L\sqrt{d_{\max}}\varepsilon_\phi + \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\,\|f_L\|_{L^4} + \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\left\|\tilde{f}_L\right\|_{L^4|\mathcal{G}_{<l}}$$

Proof. Since $\{\pm 1\}^d = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_L \cup \mathcal{G}_{\le L}$, we have

$$\left\|f_L - \tilde{f}_L\right\|_{L^2} \le \left\|\left(f_L - \tilde{f}_L\right)\mathbb{1}_{\mathcal{G}_{\le L}}\right\|_{L^2} + \sum_{l=1}^{L}\left\|\mathbb{1}_{\mathcal{B}_l} f_L\right\|_{L^2} + \sum_{l=1}^{L}\left\|\mathbb{1}_{\mathcal{B}_l}\tilde{f}_L\right\|_{L^2}.$$

Consider the first term. Let $\hat{\boldsymbol{x}}^{(l,l')}$ be the version of $\boldsymbol{x}^{(l)}$ with $\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{l'}$ replaced by $\tilde{\boldsymbol{\phi}}_1, \ldots, \tilde{\boldsymbol{\phi}}_{l'}$. Note that $\boldsymbol{x}^{(l)} = \hat{\boldsymbol{x}}^{(l,0)}$ and $\tilde{\boldsymbol{x}}^{(l)} = \hat{\boldsymbol{x}}^{(l,l-1)}$. Hence, for any $l \in [L+1]$, we can write $\tilde{\boldsymbol{x}}^{(l)} - \boldsymbol{x}^{(l)}$ as the telescoping sum

$$\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x}) - \boldsymbol{x}^{(l)}(\boldsymbol{x}) = \sum_{l'=1}^{l-1}\left(\hat{\boldsymbol{x}}^{(l,l')}(\boldsymbol{x}) - \hat{\boldsymbol{x}}^{(l,l'-1)}(\boldsymbol{x})\right).$$

In particular, this implies that for any $\boldsymbol{x} \in \{\pm 1\}^d$,

$$\left|f_L(\boldsymbol{x}) - \tilde{f}_L(\boldsymbol{x})\right| \le \left\|\boldsymbol{x}^{(L+1)}(\boldsymbol{x}) - \tilde{\boldsymbol{x}}^{(L+1)}(\boldsymbol{x})\right\| \le \sum_{l=1}^{L}\left\|\hat{\boldsymbol{x}}^{(L+1,l)}(\boldsymbol{x}) - \hat{\boldsymbol{x}}^{(L+1,l-1)}(\boldsymbol{x})\right\|.$$

Note that in $\hat{\boldsymbol{x}}^{(L+1,l)}(\boldsymbol{x}), \hat{\boldsymbol{x}}^{(L+1,l-1)}(\boldsymbol{x})$, the activations $\boldsymbol{\phi}_{l+1}, \cdots, \boldsymbol{\phi}_L$ are unchanged and all of them are C_ϕ-Lipschitz functions. Also recall that $\left\|\boldsymbol{W}^{(l')}\right\|_{\mathrm{OP}} \le C_W$ for all $l' \in [L]$. Therefore, $\hat{\boldsymbol{x}}^{(L+1,l)}, \hat{\boldsymbol{x}}^{(L+1,l-1)}$ are the same $(C_\phi C_W)^L$-Lipschitz function of their corresponding layer-l outputs. As a result, we can further rewrite the above as

$$\left|f_L(\boldsymbol{x}) - \tilde{f}_L(\boldsymbol{x})\right| \le (C_\phi C_W)^L \sum_{l=1}^{L}\left\|\boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right) - \tilde{\boldsymbol{\phi}}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right)\right\|. \tag{12}$$

Recall that $\tilde{\boldsymbol{\phi}}_l$ is chosen so that the L^∞-error is at most ε_ϕ in the approximation intervals. Therefore,

$$\left|f_L(\boldsymbol{x}) - \tilde{f}_L(\boldsymbol{x})\right| \le (C_\phi C_W)^L \sum_{l=1}^{L}\sqrt{d_{l+1}}\varepsilon_\phi \le L(C_\phi C_W)^L\sqrt{d_{\max}}\varepsilon_\phi, \quad \forall \boldsymbol{x} \in \mathcal{G}_{\le L}.$$

As a result, for the first error term, we have

$$\left\|\left(f_L - \tilde{f}_L\right)\mathbb{1}_{\mathcal{G}_{\le L}}\right\|_{L^2} \le \left\|\left(f_L - \tilde{f}_L\right)\mathbb{1}_{\mathcal{G}_{\le L}}\right\|_{L^\infty} \le L(C_\phi C_W)^L\sqrt{d_{\max}}\varepsilon_\phi.$$

Now, consider the second error term $\sum_{l=1}^{L}\left\|\mathbb{1}_{\mathcal{B}_l} f_L\right\|_{L^2}$. By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\left\|\mathbb{1}_{\mathcal{B}_l} f_L\right\|_{L^2}^2 = \mathbb{E}\left[\mathbb{1}_{\mathcal{B}_l}(\boldsymbol{x})\mathbb{1}_{\mathcal{G}_{<l}}(\boldsymbol{x})f_L^2(\boldsymbol{x})\right] &\le \mathbb{E}\left[\mathbb{1}_{\mathcal{B}_l}(\boldsymbol{x})\right]^{1/2}\mathbb{E}\left[\mathbb{1}_{\mathcal{G}_{<l}}(\boldsymbol{x})f_L^4(\boldsymbol{x})\right]^{1/2} \\ &= \mathbb{P}\left[\mathcal{B}_l\right]^{1/2}\mathbb{E}\left[\mathbb{1}_{\mathcal{G}_{<l}}(\boldsymbol{x})f_L^4(\boldsymbol{x})\right]^{1/2}.\end{aligned}$$

Therefore,

$$\sum_{l=1}^{L}\left\|\mathbb{1}_{\mathcal{B}_l} f_L\right\|_{L^2} \le \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\,\mathbb{E}\left[\mathbb{1}_{\mathcal{G}_{<l}}(\boldsymbol{x})f_L^4(\boldsymbol{x})\right]^{1/4} \le \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\left\|f_L\right\|_{L^4}.$$

Similarly, we also have

$$\begin{aligned}\sum_{l=1}^{L}\left\|\mathbb{1}_{\mathcal{B}_l}\tilde{f}_L\right\|_{L^2} \le \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\,\mathbb{E}\left[\mathbb{1}_{\mathcal{G}_{<l}}(\boldsymbol{x})\tilde{f}_L^4(\boldsymbol{x})\right]^{1/4} &\le \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\,\mathbb{E}\left[\tilde{f}_L^4(\boldsymbol{x}) \mid \mathcal{G}_{<l}\right]^{1/4} \\ &= \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\left\|\tilde{f}_L\right\|_{L^4|\mathcal{G}_{<l}}.\end{aligned}$$

Combining the above bounds completes the proof. □

By the above lemma, to control the approximation error, it suffices to bound $\mathbb{P}[\mathcal{B}_l]$ and the (conditional) fourth moments of f_L and $\tilde{f}_L$. To this end, we first bound $\mathbb{P}[\mathcal{B}_l]$ and show that, on the event $\mathcal{G}_{<l}$, $\tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)}$ is subgaussian.

Lemma B.7. *Suppose that Assumption 2.1 holds and*

$$\varepsilon_\phi \le \left(L(C_\phi C_W)^L \sqrt{d_{\max}}\right)^{-1}, \quad R_1 \ge \cdots R_L \ge 2C_W \sigma_X \sqrt{2\log(4Ld_{\max})}.$$

Then, for any $l \in [L]$ and $t \ge 0$, $\boldsymbol{v} \in \mathbb{S}^{d_l - 1}$, we have

$$\mathbb{P}[\mathcal{B}_l] \le 2d_{\max} \exp\left(-\frac{(R_l/C_W)^2}{2 \cdot 4\sigma_X^2}\right), \quad \mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle\right| \ge t \mid \mathcal{G}_{<l}\right] \le 2\exp\left(-\frac{t^2}{2 \cdot 8\sigma_X^2}\right).$$

In other words, when conditioned on $\mathcal{G}_{<l}$, $\tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)}$ has an $(8\sigma_X^2)$-subgaussian tail.

Proof. The proof of (12), *mutatis mutandis*, shows that

$$\left\|\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x}) - \boldsymbol{x}^{(l)}(\boldsymbol{x})\right\| \le (C_\phi C_W)^L \sum_{l'=1}^{l-1} \left\|\tilde{\boldsymbol{\phi}}_{l'}\left(\boldsymbol{W}^{(l')}\tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x})\right) - \boldsymbol{\phi}_{l'}\left(\boldsymbol{W}^{(l')}\tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x})\right)\right\|.$$

On event $\mathcal{G}_{<l}$, for every $l' \in [l-1]$, the entries of every $\boldsymbol{W}^{(l')}\tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x})$ lie in the corresponding approximation intervals. Therefore, we have $\left\|\tilde{\boldsymbol{\phi}}_{l'}\left(\boldsymbol{W}^{(l')}\tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x})\right) - \boldsymbol{\phi}_{l'}\left(\boldsymbol{W}^{(l')}\tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x})\right)\right\| \le \sqrt{d_{\max}}\varepsilon_\phi$. Thus,

$$\left\|\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x}) - \boldsymbol{x}^{(l)}(\boldsymbol{x})\right\| \le L(C_\phi C_W)^L \sqrt{d_{\max}}\varepsilon_\phi, \quad \forall \boldsymbol{x} \in \mathcal{G}_{<l}. \tag{13}$$

Fix an arbitrary direction $\boldsymbol{v} \in \mathbb{S}^{d_l - 1}$. The above bound implies that, for any $\boldsymbol{x} \in \mathcal{G}_{<l}$,

$$\begin{aligned}\left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle = \left\langle \boldsymbol{v}, \boldsymbol{x}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle + \left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \boldsymbol{x}^{(l)} \right\rangle &= \left\langle \boldsymbol{v}, \boldsymbol{x}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle \pm L(C_\phi C_W)^L \sqrt{d_{\max}}\varepsilon_\phi \\ &= \left\langle \boldsymbol{v}, \boldsymbol{x}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle \pm 1,\end{aligned}$$

where the second line comes from the condition $\varepsilon_\phi \le \left(L(C_\phi C_W)^L \sqrt{d_{\max}}\right)^{-1}$. Therefore, for any $t \ge 2$, we have

$$\begin{aligned}\mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle\right| \ge t \mid \mathcal{G}_{<l}\right] &\le \mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \boldsymbol{x}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle\right| \ge t/2 \mid \mathcal{G}_{<l}\right] \\ &\le \mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \boldsymbol{x}^{(l)} - \mathbb{E}\,\boldsymbol{x}^{(l)} \right\rangle\right| \ge t/2\right] / \mathbb{P}[\mathcal{G}_{<l}] \\ &\le \frac{2}{\mathbb{P}[\mathcal{G}_{<l}]} \exp\left(-\frac{t^2}{2 \cdot 4\sigma_X^2}\right),\end{aligned} \tag{14}$$

where the last line comes from (11) (or Lemma B.4). To turn this into a proper tail bound, we lower-bound the denominator $\mathbb{P}[\mathcal{G}_{<l}]$. First, by the union bound, we have

$$1 - \mathbb{P}[\mathcal{G}_{<l}] = \mathbb{P}[\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{l-1}] \le \sum_{l'=1}^{l-1} \mathbb{P}[\mathcal{B}_{l'}].$$

Furthermore, for each l', we have

$$\mathbb{P}[\mathcal{B}_{l'}] \le \sum_{k=1}^{d_{l'+1}} \mathbb{P}\left[\left|\left\langle \boldsymbol{w}_k^{(l')}, \tilde{\boldsymbol{x}}^{(l')}(\boldsymbol{x}) - \mathbb{E}\,\boldsymbol{x}^{(l')} \right\rangle\right| > R_{l'} \mid \mathcal{G}_{<l'}\right] \mathbb{P}[\mathcal{G}_{<l'}].$$

Applying (14) to each summand yields the unconditioned bound

$$\mathbb{P}[\mathcal{B}_{l'}] \le 2\sum_{k=1}^{d_{l'+1}} \exp\left(-\frac{(R_{l'}/C_W)^2}{2 \cdot 4\sigma_X^2}\right) \le 2d_{\max} \exp\left(-\frac{(R_{l'}/C_W)^2}{2 \cdot 4\sigma_X^2}\right).$$

This proves the first part of the lemma. Moreover, it implies

$$1 - \mathbb{P}[\mathcal{G}_{<l}] \le \sum_{l'=1}^{l-1} 2d_{\max} \exp\left(-\frac{(R_{l'}/C_W)^2}{2 \cdot 4\sigma_X^2}\right) \le 2Ld_{\max} \exp\left(-\frac{(R_{l-1}/C_W)^2}{2 \cdot 4\sigma_X^2}\right),$$

where the second inequality comes from $R_1 \geq \cdots \geq R_L$. In particular, the right-hand side is at most $1/2$ whenever

$$R_L \geq 2C_W \sigma_X \sqrt{2 \log (4Ld_{\max})}.$$

Combining this fact with (14) gives

$$\mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\, \boldsymbol{x}^{(l)} \right\rangle\right| \geq t \mid \mathcal{G}_{<l}\right] \leq 4 \exp\left(-\frac{t^2}{2 \cdot 4\sigma_X^2}\right), \quad \forall t \geq 2.$$

The left-hand side is trivially bounded by 1. Thus, the above implies that

$$\mathbb{P}\left[\left|\left\langle \boldsymbol{v}, \tilde{\boldsymbol{x}}^{(l)} - \mathbb{E}\, \boldsymbol{x}^{(l)} \right\rangle\right| \geq t \mid \mathcal{G}_{<l}\right] \leq 2 \exp\left(-\frac{t^2}{16\sigma_X^2}\right), \quad \forall t \geq 0.$$

Indeed, when $t^2 \leq 16(\log 2)\sigma_X^2$, the right-hand side is at least 1. When $t^2 \geq 16(\log 2)\sigma_X^2 \geq 1$, we have

$$\frac{2 \exp\left(-\frac{t^2}{16\sigma_X^2}\right)}{4 \exp\left(-\frac{t^2}{2 \cdot 4\sigma_X^2}\right)} = \frac{1}{2} \exp\left(\frac{t^2}{16\sigma_X^2}\right) \geq \frac{1}{2} \exp\left(\log 2\right) = 1.$$

□

Now, consider the L^4-norm of f_L. Because f_L uses the exact intermediate representations, we can directly apply Lemma B.4 to the last layer to control f_L.

Lemma B.8. *Suppose that Assumption 2.1 holds. Then, we have*

$$\|f_L\|_{L^4} \leq 3C_X \sqrt{d_{\max}}.$$

Proof. First, we write

$$\|f_L\|_{L^4}^4 = \mathbb{E}\left\langle \boldsymbol{a}, \boldsymbol{x}^{(L+1)} \right\rangle^4 \leq 16\, \mathbb{E}\left\langle \boldsymbol{a}, \boldsymbol{x}^{(L+1)} - \mathbb{E}\, \boldsymbol{x}^{(L+1)} \right\rangle^4 + 16\, \mathbb{E}\left\langle \boldsymbol{a}, \mathbb{E}\, \boldsymbol{x}^{(L+1)} \right\rangle^4.$$

By Lemma B.4 and Lemma B.5, $\boldsymbol{x}^{(L+1)} - \mathbb{E}\, \boldsymbol{x}^{(L+1)}$ has a σ_X^2-subgaussian tail and $\left\|\mathbb{E}\, \boldsymbol{x}^{(L+1)}\right\| \leq C_X \sqrt{d_{\max}}$. As a result, we can further bound the above expression by

$$\|f_L\|_{L^4}^4 \leq 256\sigma_X^4 + 16C_X^4 d_{\max}^2 \leq 20C_X^4 d_{\max}^2.$$

□

Finally, we consider the $(L^4|\mathcal{G}_{<l})$-norm of $\tilde{f}_L$. To this end, it suffices to control the $(L^r|\mathcal{G}_{<l'})$-norm of $\tilde{\boldsymbol{x}}^{(l)}$.

Lemma B.9. *Suppose that Assumption 2.1 and the hypotheses of Lemma B.7 hold, and $R_1 \leq \sqrt{d_{\max}}$. For any $l \in [L+1]$, $l' \in [l]$ and $r \geq 1$ with $rp_L \cdots p_1 \leq \sqrt{d_{\max}}$, there exists a universal constant $C \geq 1$ such that*

$$\left\|\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right\|_{L^r|\mathcal{G}_{<l'}} \leq \left(CC_\phi C_W C_X d_{\max}\right)^{2Lp_{l-1}\cdots p_{l'}}.$$

Proof. By Lemma B.7, when conditioned on $\mathcal{G}_{<l'}$, $\tilde{\boldsymbol{x}}^{(l')} - \mathbb{E}\, \boldsymbol{x}^{(l')}$ has an $(8\sigma_X^2)$-subgaussian tail. Consider the part of $\tilde{\boldsymbol{x}}^{(l)}$ after the l'-th layer. Recall that $\tilde{\phi}_{l'',k}$ is the degree-$p_{l''}$ Chebyshev approximation to $\phi_{l'',k}$ on $[\bar{z}_k^{(l'')} - R_{l''}, \bar{z}_k^{(l'')} + R_{l''}]$. In other words, we have

$$\tilde{\phi}_{l'',k}(z) = \tilde{\psi}_{l'',k}\left(\left(z - \bar{z}_k^{(l'')}\right)/R_{l''}\right), \quad \forall z \in \mathbb{R},\, l'' \in [L],$$

where $\tilde{\psi}_{l'',k}$ is the degree-$p_{l''}$ Chebyshev approximation to $\psi_{l'',k}(u) = \phi_{l'',k}(\bar{z}_k^{(l'')} + R_{l''}u)$ over $[-1,1]$. Hence, we can rewrite $\tilde{\boldsymbol{x}}^{(l)}$ as

$$\tilde{\boldsymbol{x}}^{(l''+1)} = \tilde{\boldsymbol{\psi}}_{l''}\left(R_{l''}^{-1}\left(\boldsymbol{W}^{(l'')}\tilde{\boldsymbol{x}}^{(l'')} - \bar{\boldsymbol{z}}^{(l'')}\right)\right), \quad \forall l'' = l', \ldots, l-1.$$

We now bound the moments of $\tilde{\boldsymbol{x}}^{(l)}$ using this recurrence relation. Fix $l'' \in \{l', \dots, l-1\}$, $r \geq 1$, and consider the $(L^r|\mathcal{G}_{<l'})$-norm of $\tilde{\boldsymbol{x}}^{(l''+1)}$. Using the fact $\|\cdot\|_2 \leq \|\cdot\|_1$, we write

$$\left\|\tilde{\boldsymbol{x}}^{(l''+1)}\right\|_{L^r|\mathcal{G}_{<l'}} \leq \sum_{k=1}^{d_{l''+1}} \left\|\tilde{x}_k^{(l''+1)}\right\|_{L^r|\mathcal{G}_{<l'}} = \sum_{k=1}^{d_{l''+1}} \mathbb{E}\,|_{\mathcal{G}_{<l'}}\left[\left|\tilde{\psi}_{l'',k}\right|^r \left(R_{l''}^{-1}\left(\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle - \bar{z}_k^{(l'')}\right)\right)\right]^{1/r}, \tag{15}$$

where $\mathbb{E}\,|_{\mathcal{G}_{<l'}}[\cdot] := \mathbb{E}[\cdot \mid \mathcal{G}_{<l'}]$. Since $\tilde{\psi}_{l'',k}$ is a polynomial, we can bound its growth using Lemma A.3, which only requires a bound on $\tilde{\psi}_{l'',k}|_{[-1,1]}$. Since $\tilde{\psi}_{l'',k}$ is chosen to approximate $\psi_{l'',k}$ to ε_ϕ-accuracy on $[-1,1]$, it suffices to bound $\psi_{l'',k}$. For any $u \in [-1,1]$, by Assumption 2.1(c), we have

$$\begin{aligned}
|\psi_{l'',k}(u)| = \left|\phi_{l'',k}(\bar{z}_k^{(l'')} + R_{l''}u)\right| &\leq |\phi_{l'',k}(0)| + C_\phi\left(\left|\bar{z}_k^{(l'')}\right| + R_{l''}\right) \\
&\leq C_\phi\left(1 + \left|\bar{z}_k^{(l'')}\right| + R_{l''}\right) \\
&\leq C_\phi\left(1 + C_W C_X\sqrt{d_{\max}} + R_{l''}\right),
\end{aligned}$$

where the last line comes from Lemma B.5 and

$$\left|\bar{z}_k^{(l'')}\right| = \left|\boldsymbol{w}_k^{(l'')} \cdot \mathbb{E}\,\boldsymbol{x}^{(l'')}\right| \leq C_W\left\|\mathbb{E}\,\boldsymbol{x}^{(l'')}\right\| \leq C_W C_X\sqrt{d_{\max}}. \tag{16}$$

Also, recall that $R_{l''} \leq R_1$. Therefore, we have

$$\left\|\tilde{\psi}_{l'',k}\right\|_{L^\infty([-1,1])} \leq 1 + \|\psi_{l'',k}\|_{L^\infty([-1,1])} \leq 2C_\phi\left(1 + C_W C_X\sqrt{d_{\max}} + R_1\right) =: C_{\mathtt{Tmp}}.$$

Since $\tilde{\psi}_{l'',k}$ is a degree-$p_{l''}$ polynomial, by Lemma A.3, this implies

$$\left|\tilde{\psi}_{l'',k}(u)\right| \leq C_{\mathtt{Tmp}}\,(1+2|u|)^{p_{l''}}, \quad \forall u \in \mathbb{R}.$$

Combining this with (15), we obtain

$$\left\|\tilde{\boldsymbol{x}}^{(l''+1)}\right\|_{L^r|\mathcal{G}_{<l'}} \leq C_{\mathtt{Tmp}}\sum_{k=1}^{d_{l''+1}} \mathbb{E}\,|_{\mathcal{G}_{<l'}}\left[\left(1 + 2\left|R_{l''}^{-1}\left(\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle - \bar{z}_k^{(l'')}\right)\right|\right)^{rp_{l''}}\right]^{1/r}.$$

For the term inside the expectation, by repeatedly using the inequality $(a+b)^p \leq 2^p(|a|^p + |b|^p)$, we obtain

$$\begin{aligned}
&\left(1 + 2\left|R_{l''}^{-1}\left(\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle - \bar{z}_k^{(l'')}\right)\right|\right)^{rp_{l''}} \\
&\leq 2^{rp_{l''}}\left(1 + 2^{rp_{l''}}\left|R_{l''}^{-1}\left(\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle - \bar{z}_k^{(l'')}\right)\right|^{rp_{l''}}\right) \\
&\leq 2^{rp_{l''}}\left(1 + (4/R_{l''})^{rp_{l''}}\left|\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle\right|^{rp_{l''}} + (4/R_{l''})^{rp_{l''}}\left|\bar{z}_k^{(l'')}\right|^{rp_{l''}}\right) \\
&\leq 2^{rp_{l''}}\left(1 + \left|\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle\right|^{rp_{l''}} + \left(C_W C_X\sqrt{d_{\max}}\right)^{rp_{l''}}\right),
\end{aligned}$$

where the last line comes from $R_{l''} \gg 1$ and (16). As a result, we have

$$\begin{aligned}
\left\|\tilde{\boldsymbol{x}}^{(l''+1)}\right\|_{L^r|\mathcal{G}_{<l'}} &\leq C_{\mathtt{Tmp}}2^{p_{l''}}\sum_{k=1}^{d_{l''+1}}\left(1 + \mathbb{E}\,|_{\mathcal{G}_{<l'}}\left[\left|\left\langle \boldsymbol{w}_k^{(l'')}, \tilde{\boldsymbol{x}}^{(l'')}\right\rangle\right|^{rp_{l''}}\right] + \left(C_W C_X\sqrt{d_{\max}}\right)^{rp_{l''}}\right)^{1/r} \\
&\leq C_{\mathtt{Tmp}}2^{p_{l''}}\sum_{k=1}^{d_{l''+1}}\left(1 + C_W^{rp_{l''}}\,\mathbb{E}\,|_{\mathcal{G}_{<l'}}\left\|\tilde{\boldsymbol{x}}^{(l'')}\right\|^{rp_{l''}} + \left(C_W C_X\sqrt{d_{\max}}\right)^{rp_{l''}}\right)^{1/r} \\
&\leq C_{\mathtt{Tmp}}d_{\max}\left(\left(4C_W C_X\sqrt{d_{\max}}\right)^{p_{l''}} + \left(2C_W\left\|\tilde{\boldsymbol{x}}^{(l'')}\right\|_{L^{rp_{l''}}|\mathcal{G}_{<l'}}\right)^{p_{l''}}\right).
\end{aligned} \tag{17}$$

We now inductively construct an upper bound $B(l''; r')$ for each $\left\|\tilde{\boldsymbol{x}}^{(l'')}\right\|_{L^{r'}|\mathcal{G}_{<l'}}$, where $l'' \in \{l', \dots, l-1\}$, $r' \in \mathbb{N}_+$. Define these bounds recursively by

$$\begin{aligned}
B(l'; r') &:= \left\|\tilde{\boldsymbol{x}}^{(l')}\right\|_{L^{r'}|\mathcal{G}_{<l'}} \vee 2C_X\sqrt{d_{\max}}, && \forall r' \in \mathbb{N}_+, \\
B(l''+1; r') &:= C_{\mathtt{Tmp}}d_{\max}(4C_W)^{p_{l''}}B(l''; r'p_{l''})^{p_{l''}}, && \forall r' \in \mathbb{N}_+,\ l'' \in \{l', \dots, l-1\}.
\end{aligned}$$

Here, the boundary condition is chosen so that $B(l';r') \geq \left\|\tilde{\boldsymbol{x}}^{(l')}\right\|_{L^{r'}|\mathcal{G}_{<l'}}$ and we can merge the $4C_W C_X\sqrt{d_{\max}}$ part into the $2C_W B(l'';rp_{l''})^{p_{l''}}$ part in (17). By (17), we have $B(l'';r') \geq \left\|\tilde{\boldsymbol{x}}^{(l'')}\right\|_{L^{r'}|\mathcal{G}_{<l'}}$. Unrolling this recurrence, we obtain

$$\begin{aligned}
B(l;r) &= C_{\texttt{Tmp}} d_{\max} (4C_W)^{p_{l-1}} B(l-1;rp_{l-1})^{p_{l-1}} \\
&= (C_{\texttt{Tmp}} d_{\max})^{1+p_{l-1}} (4C_W)^{p_{l-1}+p_{l-1}p_{l-2}} B(l-2;rp_{l-1}p_{l-2})^{p_{l-1}p_{l-2}} \\
&= (C_{\texttt{Tmp}} d_{\max})^{1+p_{l-1}+\cdots+p_{l-1}\cdots p_{l'+1}} (4C_W)^{p_{l-1}+\cdots+p_{l-1}\cdots p_{l'}} B(l';rp_{l-1}\cdots p_{l'})^{p_{l-1}\cdots p_{l'}} \\
&\leq (4C_W C_{\texttt{Tmp}} d_{\max})^{Lp_{l-1}\cdots p_{l'}} B(l';rp_{l-1}\cdots p_{l'})^{p_{l-1}\cdots p_{l'}}.
\end{aligned}$$

To upper bound $B(l';rp_{l-1}\cdots p_{l'})$, recall that $\left\|\mathbb{E}\,\boldsymbol{x}^{(l')}\right\| \leq C_X\sqrt{d_{\max}}$ by Lemma B.5, and, by Lemma B.7, when conditioned on $\mathcal{G}_{<l'}$, $\tilde{\boldsymbol{x}}^{(l')} - \mathbb{E}\,\boldsymbol{x}^{(l')}$ has an $(8\sigma_X^2)$-subgaussian tail. Hence,

$$\begin{aligned}
\left\|\tilde{\boldsymbol{x}}^{(l')}\right\|_{L^{rp_{l-1}\cdots p_{l'}}|\mathcal{G}_{<l'}} &\leq \left\|\tilde{\boldsymbol{x}}^{(l')} - \mathbb{E}\,\boldsymbol{x}^{(l')}\right\|_{L^{rp_{l-1}\cdots p_{l'}}|\mathcal{G}_{<l'}} + \left\|\mathbb{E}\,\boldsymbol{x}^{(l')}\right\| \\
&\lesssim \sigma_X\sqrt{d_{\max}} + \sigma_X\sqrt{rp_{l-1}\cdots p_{l'}} + C_X\sqrt{d_{\max}} \\
&\lesssim C_X\sqrt{d_{\max}},
\end{aligned}$$

where the last line comes from the condition $rp_L\cdots p_1 \leq \sqrt{d_{\max}}$. This implies that

$$B(l';rp_{l-1}\cdots p_{l'}) = \left\|\tilde{\boldsymbol{x}}^{(l')}\right\|_{L^{rp_{l-1}\cdots p_{l'}}|\mathcal{G}_{<l'}} \vee 2C_X\sqrt{d_{\max}} \leq CC_X\sqrt{d_{\max}},$$

for some universal constant $C \geq 1$. Thus, we have

$$B(l;r) \leq (4C_W C_{\texttt{Tmp}} d_{\max})^{Lp_{l-1}\cdots p_{l'}} \left(CC_X\sqrt{d_{\max}}\right)^{p_{l-1}\cdots p_{l'}} \leq \left(CC_\phi C_W^2 C_X^2 d_{\max}^2\right)^{Lp_{l-1}\cdots p_{l'}}.$$

□

Corollary B.10. *Under the conditions of Lemma B.9, there is a universal constant $C \geq 1$ such that*

$$\left\|\tilde{f}_L\right\|_{L^4|\mathcal{G}_{<l}} \leq \left(CC_\phi C_W C_X d_{\max}\right)^{2Lp_L\cdots p_l}, \quad \forall l \in [L+1].$$

Proof. By Lemma B.9, we have

$$\left\|\tilde{\boldsymbol{x}}^{(L+1)}(\boldsymbol{x})\right\|_{L^r|\mathcal{G}_{<l}} \leq \left(CC_\phi C_W C_X d_{\max}\right)^{2Lp_L\cdots p_l}, \quad \forall l \in [L+1].$$

Therefore,

$$\left\|\tilde{f}_L\right\|_{L^4|\mathcal{G}_{<l}} = \left\|\left\langle \boldsymbol{a}, \tilde{\boldsymbol{x}}^{(L+1)}\right\rangle\right\|_{L^4|\mathcal{G}_{<l}} \leq \left\|\tilde{\boldsymbol{x}}^{(L+1)}\right\|_{L^4|\mathcal{G}_{<l}} \leq \left(CC_\phi C_W C_X d_{\max}\right)^{2Lp_L\cdots p_l}.$$

□

We now combine the above bounds to prove Lemma 2.2. For convenience, we restate it below.

Lemma 2.2. *Suppose that Assumption 2.1 holds. Then, for any $\varepsilon_* \geq d^{-O(1)}$, there exists a polynomial $\tilde{f}_L : \{\pm 1\}^d \to \mathbb{R}$ such that $\left\|f_L - \tilde{f}_L\right\|_{L^2} \leq \varepsilon_*$ and*

$$\deg \tilde{f}_L \lesssim \left((C_\phi C_W^2)^{2^{L+3}} (\log d)^{2L+2}\right)^{L(2^L+1)} \ll d^{1/2}.$$

Proof. First, recall from Lemma B.6 that

$$\left\|f_L - \tilde{f}_L\right\|_{L^2} \leq L(C_\phi C_W)^L\sqrt{d_{\max}}\varepsilon_\phi + \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\left\|f_L\right\|_{L^4} + \sum_{l=1}^{L}\mathbb{P}[\mathcal{B}_l]^{1/4}\left\|\tilde{f}_L\right\|_{L^4|\mathcal{G}_{<l}}.$$

By Lemma B.7, Lemma B.8, and Corollary B.10, we can further bound the right-hand side by

$$\begin{aligned}\left\|f_L - \tilde{f}_L\right\|_{L^2} &\le L(C_\phi C_W)^L\sqrt{d_{\max}}\varepsilon_\phi + \sum_{l=1}^{L} 2d_{\max}\exp\left(-\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2}\right)3C_X\sqrt{d_{\max}} \\ &\quad + \sum_{l=1}^{L} 2d_{\max}\exp\left(-\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2}\right)(CC_\phi C_W C_X d_{\max})^{2Lp_L\cdots p_l} \\ &=: \mathtt{Tmp}_1 + \mathtt{Tmp}_2 + \mathtt{Tmp}_3.\end{aligned}$$

For the first term to be smaller than $\varepsilon_*/3$, it suffices to choose

$$\varepsilon_\phi := \frac{\varepsilon_*}{3L(C_\phi C_W)^L\sqrt{d_{\max}}}. \tag{18}$$

For the second term, since $R_L \le \cdots \le R_1$, we have

$$\mathtt{Tmp}_2 \le L\cdot 2d_{\max}\exp\left(-\frac{(R_L/C_W)^2}{8\cdot 4\sigma_X^2}\right)3C_X\sqrt{d_{\max}}.$$

Thus, for $\mathtt{Tmp}_2$ to be bounded by $\varepsilon_*/3$, it suffices to require

$$R_L \ge 20C_W\sigma_X\sqrt{\log\left(LC_X d_{\max}/\varepsilon_*\right)}. \tag{19}$$

Finally, consider the third error term. For each $l \in [L]$, by Assumption 2.1(d) and Lemma A.6, to achieve ε_ϕ-accuracy over $[\bar{z}_k^{(l)} - R_l, \bar{z}_k^{(l)} + R_l]$, it suffices to choose

$$\frac{4R_l}{c_\phi}\log\left(\frac{4R_l}{c_\phi\varepsilon_\phi}C_\phi\left(1 + \left|\bar{z}_k^{(l)}\right| + R_l + c_\phi/2\right)^{m_\phi}\right) \le C_1 R_l\log(d_{\max}/\varepsilon_\phi) =: p_l,$$

where $C_1 \ge 1$ is a large universal constant and ε_ϕ is given by (18). We then bound $\mathtt{Tmp}_3$ by

$$\begin{aligned}\mathtt{Tmp}_3 &\le L\max_{l\in[L]} 2d_{\max}\exp\left(-\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2}\right)(CC_\phi C_W C_X d_{\max})^{2Lp_L\cdots p_l} \\ &\le 2Ld_{\max}\max_{l\in[L]}\exp\left(-\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2}\right)(CC_\phi C_W C_X d_{\max})^{2L(C_1\log(d_{\max}/\varepsilon_\phi))^L R_L\cdots R_l} \\ &=: 2Ld_{\max}\max_{l\in[L]}\mathtt{Tmp}_{3,l}.\end{aligned}$$

Now, we choose $(R_l)_l$ so that $\mathtt{Tmp}_{3,l} \le \varepsilon_*/(6Ld_{\max})$ for all $l \in [L]$. For notational simplicity, define

$$\Lambda := 2L(C_1\log(d_{\max}/\varepsilon_\phi))^L\log(CC_\phi C_W C_X d_{\max}). \tag{20}$$

Then, we can rewrite $\mathtt{Tmp}_{3,l}$ as

$$\mathtt{Tmp}_{3,l} = \exp\left(-\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2} + \Lambda R_L\cdots R_l\right), \quad \forall l \in [L].$$

Suppose that $R_L, \cdots, R_{l+1}$ have been chosen. Then, for $\mathtt{Tmp}_{3,l}$ to be bounded by $\varepsilon_*/(6Ld_{\max})$, it suffices to require

$$\begin{aligned}&\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2} \ge 2\Lambda R_L\cdots R_l, \quad \exp\left(-\frac{1}{2}\frac{(R_l/C_W)^2}{8\cdot 4\sigma_X^2}\right) \le \frac{\varepsilon_*}{6Ld_{\max}}, \\ &\qquad \Leftarrow \quad R_l \ge 64C_W^2\sigma_X^2\Lambda R_L\cdots R_{l+1}, \quad R_l \ge 8C_W\sigma_X\sqrt{\log\left(6Ld_{\max}/\varepsilon_*\right)}.\end{aligned}$$

Since we require $R_L \le \cdots \le R_1$, we can replace R_l with R_L in the second condition. Combining this with (19) yields the following backward recurrence:

$$\begin{aligned}&R_L \ge 20C_W\sigma_X\sqrt{\log\left(LC_X d/\varepsilon_*\right)} \vee 64C_W^2\sigma_X^2\Lambda \vee 8C_W\sigma_X\sqrt{\log\left(6Ld_{\max}/\varepsilon_*\right)}, \\ &R_l \ge 64C_W^2\sigma_X^2\Lambda R_L\cdots R_{l+1}.\end{aligned}$$

Recall the definition of Λ from (20). To satisfy the condition on R_L, it suffices to choose

$$R_L = C_2 C_W^2\sigma_X^2\log^{L+1}(d_{\max}/\varepsilon_\phi), \tag{21}$$

for some large universal constant $C_2 \geq 1$. Similarly, we can simplify the recurrence relation as

$$R_l \geq C_3 C_W^2 \sigma_X^2 \log^{L+1}(d_{\max}/\varepsilon_\phi) R_L \cdots R_{l+1},$$

where $C_3 \geq 1$ is a large universal constant. Solving this recurrence relation gives

$$R_l = \left(C_3 C_W^2 \sigma_X^2 \log^{L+1}(d_{\max}/\varepsilon_\phi) \cdot R_L\right)^{2^{L-l-1}}, \quad \forall l \in [L-1]. \tag{22}$$

In particular, this implies that for every $l \in [L]$,

$$\begin{aligned} p_l = C_1 R_l \log(d_{\max}/\varepsilon_\phi) &\leq C_1 \left(C_3 C_W^2 \sigma_X^2 \log^{L+1}(d_{\max}/\varepsilon_\phi) \cdot R_L\right)^{2^L} \log(d_{\max}/\varepsilon_\phi) \\ &\leq C_4 \left(C_W^4 \sigma_X^4 \log^{2L+2}(d_{\max}/\varepsilon_\phi)\right)^{2^L+1}. \end{aligned}$$

Thus, when $\varepsilon_\phi \geq d^{-O(1)}$ and $d_{\max} \leq d^{O(1)}$, the total degree of $\tilde{f}$ is bounded by

$$\prod_{l=1}^{L} p_l \leq C_5 \left(C_W^4 \sigma_X^4 (\log d)^{2L+2}\right)^{L(2^L+1)} \leq C_6 \left((C_\phi C_W^2)^{2^{L+3}} (\log d)^{2L+2}\right)^{L(2^L+1)},$$

where $C_4, C_5, C_6 \geq 1$ are universal constants. Recall from Assumption 2.1 that $C_\phi, C_W \leq d^{c_0}$ with $c_0 < 2^{-2L-7}/L$. Hence, the above $(p_l)_l$ meet the condition $p_L \cdots p_1 \ll d^{1/2}$ required by Lemma B.9. □

B.3 Proof of the main lower bound theorem

Proof of Theorem 2.1. By Lemma 2.2, we can approximate f_L to accuracy $1/d$ by a polynomial $\tilde{f}_L$ whose degree is much smaller than $d^{1/2}$. Thus, by the orthogonality of Boolean monomials, we have

$$\|f_* - f_L\|_{L^2} \geq \left\|f_* - \tilde{f}_L\right\|_{L^2} - \left\|f_L - \tilde{f}_L\right\|_{L^2} \geq \left\|\mathbf{\Pi}_{>\deg(\tilde{f}_L)} f_*\right\|_{L^2} - (1/d) \geq 1/4.$$

□

C Upper bound proofs

As we have discussed in Section 3, to prove Theorem 3.1, it suffices to show by induction that, for each layer $l \in [L]$, $\boldsymbol{x}^{(l)}$ contains the features present in $\mathcal{K}_l \setminus \{\varnothing\}$. Formally, we maintain the induction hypothesis Assumption 3.1, which we restate below for convenience.

Assumption 3.1 (Induction hypothesis on the input)**.** *Fix $l \in [L]$. We assume as an induction hypothesis the following properties of the l-th-layer input $\boldsymbol{x}^{(l)} := (\boldsymbol{x}, \boldsymbol{y}^{(l-1)}) \in \mathbb{R}^{d+m}$.*

(a) For every $k \in [d+m]$, $x_k^{(l)}$, as a function of the input $\boldsymbol{x}$, is either identically zero or nowhere zero. We write $I_l := \left\{k \in [d+m] : x_k^{(l)} \not\equiv 0\right\}$ for the collection of nonzero coordinates.

(b) There exist nonzero scalars $(\alpha_k^{(l)})_{k \in I_l}$ and subsets $(S_k^{(l)})_{k \in I_l}$ of $[d]$ such that $x_k^{(l)} = \alpha_k^{(l)} \chi_{S_k^{(l)}}(\boldsymbol{x})$ with $\alpha_k^{(l)} \in \{1\} \cup (1 \pm 0.2)\hat{f}_*(S_k^{(l)})$ for every $k \in I_l$. We call $\mathcal{F}_l := \left\{S_k^{(l)}\right\}_{k \in I_l}$ the features contained in $\boldsymbol{x}^{(l)}$. Moreover, for any $S \in \mathcal{F}_l$ with $|S| \geq 2$, there is exactly one $k \in I_l$ with $S_k^{(l)} = S$.

(c) $\boldsymbol{x}^{(l)}$ contains all nonempty features in $\mathcal{K}_l$, i.e., $\mathcal{F}_l \supset \mathcal{K}_l \setminus \{\varnothing\}$.

Fact C.1 (Base case)**.** *For $l = 1$, Assumption 3.1 holds with $I_1 = [d]$, and $\alpha_k^{(1)} = 1$, $S_k^{(1)} = \{k\}$ for all $k \in [d]$.*

Consider the intermediate feature vector $\boldsymbol{z}^{(l)} := \mathbf{\Phi}(\boldsymbol{x}^{(l)})$, where $\mathbf{\Phi}$ is the degree-D polynomial feature mapping defined in (4). We index it using $H \in \binom{[d+m]}{\leq D} \simeq [M]$. We also fix an arbitrary

ordering of $\binom{[d+m]}{\leq D}$ in the inner loop of Algorithm 1 (line 5), and write $H < H'$ if H appears before H' in this order. Also, recall from (9) that $\mathcal{H}_l$ denotes the collection of nontrivial entries of $\boldsymbol{z}^{(l)}$.

To implement the proof sketch in Section 3, we start with a lemma that isolates the relevant terms in the loss (6).

Lemma C.2 (Loss decomposition)**.** *Suppose that Assumption 3.1 holds at layer l. Consider the lth iteration of the outer for loop of Algorithm 1. Fix a nonzero coordinate $H \in [M]$ of $\boldsymbol{z}^{(l)}$. Let $\mathcal{C}_H := \{H' \in \mathcal{H}_l : S_H^{(l)} = S_{H'}^{(l)}\} \subset \mathcal{H}_l$ denote the collection of the coordinates of $\boldsymbol{z}^{(l)}$ that represent the same feature as H. Then, we can decompose the loss (6) as*

$$\mathcal{L}^{(l)}\big(\boldsymbol{w}^{(l)}\big) = \frac{1}{2}\left(\hat{f}_*\big(S_H^{(l)}\big) - \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)}\right)^2 - \alpha_H^{(l)} w_H^{(l)} \mathcal{E}_H^{(l)} + \Big(w_H^{(l)}\text{-}\textit{independent terms}\Big),$$

where

$$\mathcal{E}_H^{(l)} := \sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \hat{f}_*(S) \hat{\mathbb{E}}^{(l)} \left[\chi_{S_H^{(l)}}(\boldsymbol{x}) \chi_S(\boldsymbol{x})\right] - \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \alpha_G^{(l)} w_G^{(l)} \hat{\mathbb{E}}^{(l)} \left[\chi_{S_H^{(l)}}(\boldsymbol{x}) \chi_{S_G^{(l)}}(\boldsymbol{x})\right].$$

Remark. Note that the first term in the decomposition depends only on $(w_{H'}^{(l)})_{H' \in \mathcal{C}_H}$ and is a simple quadratic function in $w_H^{(l)}$. Moreover, it does not depend on the sampled dataset. The only (relevant) source of randomness comes from the $\mathcal{E}_H^{(l)}$ part of the second term. By the orthogonality of Boolean monomials, the $\hat{\mathbb{E}}^{(l)}$ parts in $\mathcal{E}_H^{(l)}$ have expectation zero. We will show in the next lemma that when the number of samples is reasonably large, $\mathcal{E}_H^{(l)}$ is uniformly small with high probability for all $\boldsymbol{w}^{(l)}$ with $\|\boldsymbol{w}^{(l)}\|_\infty$ bounded. ♣

Proof. First, using Assumption 3.1 and (10), we can rewrite $f^{(l)}(\boldsymbol{x}) = \langle \boldsymbol{w}^{(l)}, \boldsymbol{z}^{(l)} \rangle$ as

$$f^{(l)}(\boldsymbol{x}) = \sum_{G \in \mathcal{H}_l} \alpha_G^{(l)} w_G^{(l)} \chi_{S_G^{(l)}}(\boldsymbol{x}) = \chi_{S_H^{(l)}}(\boldsymbol{x}) \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)} + \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \alpha_G^{(l)} w_G^{(l)} \chi_{S_G^{(l)}}(\boldsymbol{x}).$$

Similarly, we can rewrite the target function as

$$f_* = \hat{f}_*\big(S_H^{(l)}\big) \chi_{S_H^{(l)}} + \sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \hat{f}_*(S) \chi_S.$$

By the definition of $\mathcal{C}_H$, $S_H^{(l)}$ differs from every $S_G^{(l)}$ and S appearing in the corresponding sums, and $H \neq G$. Then, we decompose the loss as

$$\begin{aligned}
\mathcal{L}^{(l)}\big(\boldsymbol{w}^{(l)}\big) &= \frac{1}{2}\hat{\mathbb{E}}^{(l)} \left(f_*(\boldsymbol{x}) - \chi_{S_H^{(l)}}(\boldsymbol{x}) \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)} - \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \alpha_G^{(l)} w_G^{(l)} \chi_{S_G^{(l)}}(\boldsymbol{x})\right)^2 \\
&= \frac{1}{2}\hat{\mathbb{E}}^{(l)} \left(f_*(\boldsymbol{x}) - \chi_{S_H^{(l)}}(\boldsymbol{x}) \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)}\right)^2 \\
&\quad + \alpha_H^{(l)} w_H^{(l)} \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \alpha_G^{(l)} w_G^{(l)} \hat{\mathbb{E}}^{(l)} \left[\chi_{S_H^{(l)}}(\boldsymbol{x}) \chi_{S_G^{(l)}}(\boldsymbol{x})\right] + \Big(w_H^{(l)}\text{-independent terms}\Big).
\end{aligned}$$

For the first term, we can further decompose it as

$$\begin{aligned}
&\frac{1}{2}\hat{\mathbb{E}}^{(l)} \left(f_*(\boldsymbol{x}) - \chi_{S_H^{(l)}}(\boldsymbol{x}) \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)}\right)^2 \\
&\quad = \frac{1}{2}\left(\hat{f}_*\big(S_H^{(l)}\big) - \sum_{H' \in \mathcal{C}_H} \alpha_{H'}^{(l)} w_{H'}^{(l)}\right)^2 - \alpha_H^{(l)} w_H^{(l)} \sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \hat{f}_*(S) \hat{\mathbb{E}}^{(l)} \left[\chi_{S_H^{(l)}}(\boldsymbol{x}) \chi_S(\boldsymbol{x})\right] \\
&\qquad\qquad + \Big(w_H^{(l)}\text{-independent terms}\Big).
\end{aligned}$$

Combining the preceding two displays completes the proof. □

We now control the error term $\mathcal{E}_H^{(l)}$ in the above decomposition.

Lemma C.3 (Error term concentration)**.** *Let $\varepsilon_{\mathcal{E}}, \delta_{\mathbb{P}} \in (0,1)$ be the target accuracy and failure probability, respectively. Let $B_w \geq 1$ be a parameter controlling the largest possible $\left\|\boldsymbol{w}^{(l)}\right\|_\infty$. Suppose that*

$$N \gtrsim \frac{B_w^2 M^2}{\varepsilon_{\mathcal{E}}^2} \log\left(\frac{M}{\delta_{\mathbb{P}}}\right).$$

Then, with probability at least $1-\delta_{\mathbb{P}}$, $\left|\mathcal{E}_H^{(l)}\right| \leq \varepsilon_{\mathcal{E}}$ holds uniformly for all $\boldsymbol{w}^{(l)}$ with $\left\|\boldsymbol{w}^{(l)}\right\|_\infty \leq B_w$ and $H \in \mathcal{H}_l$.

Proof. Recall from Lemma C.2 that

$$\mathcal{E}_H^{(l)} := \sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \hat{f}_*(S) \hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_S(\boldsymbol{x})\right] - \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \alpha_G^{(l)} w_G^{(l)} \hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_{S_G^{(l)}}(\boldsymbol{x})\right].$$

By definition, $S_H^{(l)} \neq S$ for any $S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}$, and $S_H^{(l)} \neq S_G^{(l)}$ for any $G \in \mathcal{H}_l \setminus \mathcal{C}_H$.

Now, consider two arbitrary distinct subsets $S, S' \subset [d]$ and the random variable $\chi_S(\boldsymbol{x})\chi_{S'}(\boldsymbol{x})$. Note that $\chi_S(\boldsymbol{x})\chi_{S'}(\boldsymbol{x}) \sim \mathrm{Unif}(\{\pm 1\})$. Therefore, by Hoeffding's inequality, $\left|\hat{\mathbb{E}}_N\left[\chi_S(\boldsymbol{x})\chi_{S'}(\boldsymbol{x})\right]\right| \leq \varepsilon$, with probability at least $1-\delta_{\mathbb{P}}$, if $N \geq 2\log(2/\delta_{\mathbb{P}})/\varepsilon^2$.

There are at most Mm choices of $(H,S) \in [M] \times \mathcal{S}_*$ and M^2 choices of $(H,G) \in [M] \times [M]$. Thus, the union bound implies that

$$\left|\hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_S(\boldsymbol{x})\right]\right| \vee \left|\hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_{S_G^{(l)}}(\boldsymbol{x})\right]\right| \leq \varepsilon, \quad \forall H \in \mathcal{H}_l,\ S \in \mathcal{S}_* \setminus \{S_H^{(l)}\},\ G \in \mathcal{H}_l \setminus \mathcal{C}_H,$$

with probability at least $1 - 2M^2\delta_{\mathbb{P}}$, as long as $N \geq 2\log(2/\delta_{\mathbb{P}})/\varepsilon^2$.

Condition on this event. For any $\boldsymbol{w}^{(l)}$ with $\left\|\boldsymbol{w}^{(l)}\right\|_\infty \leq B_w$, by the Cauchy-Schwarz inequality, Parseval's theorem and Assumption 3.1(b), we have

$$\begin{aligned}
\left|\mathcal{E}_H^{(l)}\right| &\leq \sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \left|\hat{f}_*(S)\right| \left|\hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_S(\boldsymbol{x})\right]\right| + \sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \left|\alpha_G^{(l)} w_G^{(l)}\right| \left|\hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_{S_G^{(l)}}(\boldsymbol{x})\right]\right| \\
&\leq \left\|\hat{f}_*\right\| \sqrt{\sum_{S \in \mathcal{S}_* \setminus \{S_H^{(l)}\}} \hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_S(\boldsymbol{x})\right]^2} + B_w \left\|\boldsymbol{\alpha}^{(l)}\right\| \sqrt{\sum_{G \in \mathcal{H}_l \setminus \mathcal{C}_H} \hat{\mathbb{E}}^{(l)}\left[\chi_{S_H^{(l)}}(\boldsymbol{x})\chi_{S_G^{(l)}}(\boldsymbol{x})\right]^2} \\
&\leq \sqrt{|\mathcal{S}_*|}\varepsilon + B_w \left\|\boldsymbol{\alpha}^{(l)}\right\| \sqrt{M}\varepsilon \leq \left(1 + B_w \left\|\boldsymbol{\alpha}^{(l)}\right\|\right) \sqrt{M}\varepsilon.
\end{aligned}$$

By Assumption 3.1(b) and the condition $\|f_*\|_{L^2} \leq 1$, we have $\left\|\boldsymbol{\alpha}^{(l)}\right\| \leq 2^D\sqrt{M}$. Hence, we can further rewrite the above as $\left|\mathcal{E}_H^{(l)}\right| \leq 2^{D+1} B_w M \varepsilon$. Replacing $\delta_{\mathbb{P}}$ with $\delta_{\mathbb{P}}/(2M^2)$ and ε with $\varepsilon_{\mathcal{E}}/\left(2^{D+1}B_w M\right)$ completes the proof. □

A standard convex optimization argument gives the following one-dimensional quadratic optimization lemma.

Lemma C.4. *Fix arbitrary $a > 0$ and $b, c \in \mathbb{R}$ and define $\mathcal{L}(w) := aw^2 + bw + c$. Consider the update rule $w_{t+1} = w_t - \eta \mathcal{L}'(w_t)$ with $w_0 = 0$ and step size $\eta > 0$. If $\eta \leq 1/(2a)$, then $|w_t - w_*| \leq \exp(-2a\eta t)|w_*|$, where $w_* = -b/(2a)$ is the unique minimizer of $\mathcal{L}$.*

Now, we combine the above three lemmas to show that one-pass coordinate descent will learn and deduplicate the correct features.

Lemma C.5 (One-pass coordinate descent)**.** *Suppose that Assumption 3.1 holds at layer l. Consider the lth iteration of the outer for loop of Algorithm 1 (line 3). Let $\delta_{\mathbb{P}} \in (0,1)$ denote the target failure probability. Choose the algorithm parameters such that*

$$\lambda_w = 2^{-D-1}\kappa, \quad N \gtrsim \frac{M^2 \log(M/\delta_{\mathbb{P}})}{\kappa^{4D+4}}, \quad \eta \leq 2^{-2D}, \quad T \gtrsim \frac{D\log(1/\kappa)}{\kappa^{2D}\eta}.$$

Then, with probability at least $1-\delta_{\mathbb{P}}$, for every $H \in \mathcal{H}_l$, the following holds after the H-th iteration of the inner for loop (line 5):

(a) If $\hat{f}_*(S_H^{(l)}) = 0$ or there exists some $H' \in \mathcal{H}_l$ with $H' < H$ and $S_H^{(l)} = S_{H'}^{(l)}$, then $w_H^{(l)} = 0$.

(b) Otherwise, $w_H^{(l)} = (1 \pm 0.2)\hat{f}_*(S_H^{(l)})/\alpha_H^{(l)}$.

Proof. Let $\varepsilon_1, \varepsilon_2 > 0$ be parameters to be determined later. In addition, choose $N^{(l)}$ according to Lemma C.3, so that $|\mathcal{E}_H^{(l)}| \le \varepsilon_{\mathcal{E}} := \varepsilon_2 \kappa$ uniformly for all $H \in \mathcal{H}_l$.

First consider case (b); that is, H is the first among $\mathcal{C}_H$ to appear in the inner for loop (line 5) of Algorithm 1. Since we initialize $\boldsymbol{w}^{(l)} = 0$, in this case, we can further simplify Lemma C.2 as

$$\begin{aligned}\mathcal{L}^{(l)}(\boldsymbol{w}^{(l)}) &= \frac{1}{2}\left(\hat{f}_*(S_H^{(l)}) - \alpha_H^{(l)} w_H^{(l)}\right)^2 - \alpha_H^{(l)} w_H^{(l)} \mathcal{E}_H^{(l)} + \left(w_H^{(l)}\text{-independent terms}\right) \\ &= \frac{1}{2}\left(\alpha_H^{(l)}\right)^2 \left(w_H^{(l)}\right)^2 - \left(\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}\right)\alpha_H^{(l)} w_H^{(l)} + \left(w_H^{(l)}\text{-independent terms}\right).\end{aligned}$$

Recall that $\alpha_H^{(l)} := \prod_{i \in H} \alpha_i^{(l)}$ and H is a subset of $[d+m]$ of size at most D. By Assumption 3.1(b) and the condition $\|f_*\|_{L^2} \le 1$, we have $|\alpha_H^{(l)}| \le 1.2^D$. Therefore, by Lemma C.4, as long as we choose $\eta \le 2^{-2D}$, we will have, for any $t \ge 0$,

$$\begin{aligned}\left|w_H^{(l)}(t) - \frac{\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}}\right| &\le \exp\left(-\left(\alpha_H^{(l)}\right)^2 \eta t\right)\left|\frac{\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}}\right| \\ &\le \exp\left(-(\kappa/2)^{2D}\eta t\right)\left|\frac{\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}}\right|,\end{aligned}$$

where the second inequality comes from Assumption 3.1(b) and the condition that f_* has signal strength κ. As a result, for any $\varepsilon_1 > 0$, we have

$$T \ge \frac{\log(1/\varepsilon_1)}{(\kappa/2)^{2D}\eta} \quad \Rightarrow \quad w_H^{(l)}(T) = (1 \pm \varepsilon_1)\frac{\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}} = (1 \pm 2\varepsilon_1 \pm 2\varepsilon_2)\frac{\hat{f}_*(S_H^{(l)})}{\alpha_H^{(l)}}.$$

In particular, when $\varepsilon_1, \varepsilon_2 \le 0.05$, this implies

$$\left|w_H^{(l)}(T)\right| = \left|(1 \pm 0.2)\frac{\hat{f}_*(S_H^{(l)})}{\alpha_H^{(l)}}\right| \ge 0.8\frac{\kappa}{1.2^D} \ge 2^{-D}\kappa.$$

Hence, in line 7 of Algorithm 1, as long as we choose the truncation threshold λ_w to be at most $2^{-D-1}\kappa$, we will not accidentally truncate $w_H^{(l)}(T)$ to 0.

Now, consider case (a). First, suppose that there exists $H' < H$ with $S_{H'}^{(l)} = S_H^{(l)}$. Let H' be the earliest such coordinate. Then, by case (b), after the H'-th iteration, we have $w_{H'}^{(l)}(t) = (1 \pm \varepsilon_1 \pm \varepsilon_2)\hat{f}_*(S_H^{(l)})/\alpha_{H'}^{(l)}$. Therefore, the loss decomposition in Lemma C.2 becomes

$$\begin{aligned}\mathcal{L}^{(l)}(\boldsymbol{w}^{(l)}) = \frac{1}{2}\Bigg((\pm\varepsilon_1 \pm \varepsilon_2)\hat{f}_*(S_H^{(l)}) - \alpha_H^{(l)} w_H^{(l)} - \sum_{H'' \in \mathcal{C}_H \setminus \{H, H'\}} \alpha_{H''}^{(l)} w_{H''}^{(l)}\Bigg)^2 \\ - \alpha_H^{(l)} w_H^{(l)} \mathcal{E}_H^{(l)} + \left(w_H^{(l)}\text{-independent terms}\right).\end{aligned}$$

Note that this bound is also valid when $\hat{f}_*(S_H^{(l)}) = 0$. Since we initialize $\boldsymbol{w}^{(l)} = 0$, $w_{H''}^{(l)} = 0$ for all $H'' > H$. Assume as an induction hypothesis that $w_{H''}^{(l)} = 0$ for all $H'' \in \mathcal{C}_H$ between H' and H. Then, the above becomes

$$\begin{aligned}\mathcal{L}^{(l)}(\boldsymbol{w}^{(l)}) &= \frac{1}{2}\left((\pm\varepsilon_1 \pm \varepsilon_2)\hat{f}_*(S_H^{(l)}) - \alpha_H^{(l)} w_H^{(l)}\right)^2 - \alpha_H^{(l)} w_H^{(l)} \mathcal{E}_H^{(l)} + \left(w_H^{(l)}\text{-independent terms}\right) \\ &= \frac{1}{2}\left(\alpha_H^{(l)}\right)^2 \left(w_H^{(l)}\right)^2 - \left((\pm\varepsilon_1 \pm \varepsilon_2)\hat{f}_*(S_H^{(l)}) + \mathcal{E}_H^{(l)}\right)\alpha_H^{(l)} w_H^{(l)} + \left(w_H^{(l)}\text{-independent terms}\right).\end{aligned}$$

By the same analysis, we know

$$\left|w_H^{(l)}(T)\right| = \left|(1 \pm \varepsilon_1)\frac{(\pm\varepsilon_1 \pm \varepsilon_2)\hat{f}_*\big(S_H^{(l)}\big) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}}\right| \le \left|\frac{4(\varepsilon_1+\varepsilon_2)}{\alpha_H^{(l)}}\right| \le \frac{2^{D+2}(\varepsilon_1+\varepsilon_2)}{\kappa^D}.$$

This implies that in order for line 7 to truncate $w_H^{(l)}(T)$ to 0 when $\lambda_w = 2^{-D-1}\kappa$, it suffices to require

$$\frac{2^{D+2}(\varepsilon_1+\varepsilon_2)}{\kappa^D} \le 2^{-D-1}\kappa \quad \Leftarrow \quad \varepsilon_1+\varepsilon_2 \le 2^{-2D-3}\kappa^{D+1}.$$

Note that this constraint is stronger than the previous constraint imposed by case (b), and the same argument remains valid for $\hat{f}_*(S_H^{(l)}) = 0$. Hence, it suffices to keep this constraint and choose $\varepsilon_1 = \varepsilon_2 = 2^{-2D-4}\kappa^{D+1}$. In addition, since $w_H^{(l)}$ always moves toward the corresponding minimizer (cf. Lemma C.4) and we initialize it with 0, we also have

$$\left\|\boldsymbol{w}^{(l)}\right\|_\infty \le \max_H \left|\frac{\hat{f}_*\big(S_H^{(l)}\big) + \mathcal{E}_H^{(l)}}{\alpha_H^{(l)}}\right| \le 2 \cdot \frac{2^D}{\kappa^D}.$$

Finally, using Lemma C.3 and Lemma C.4, we can translate the conditions on $\varepsilon_1, \varepsilon_2$ into the following constraint on the number of samples and steps:

$$N \gtrsim \frac{M^2 \log(M/\delta_{\mathbb{P}})}{\kappa^{4D+4}}, \quad T \gtrsim \frac{D\log(1/\kappa)}{\kappa^{2D}\eta}.$$

□

With this lemma, we can now show that the induction hypothesis Assumption 3.1 holds and estimate the corresponding α's.

Lemma C.6 (Induction step)**.** *Suppose that Assumption 3.1 holds at layer $l \in [L]$ and choose the algorithm parameters according to Lemma C.5. Then, Assumption 3.1 also holds at layer $l+1$.*

Proof. Recall from the discussion after Fact C.1 that $\mathcal{F}_{l+1/2} := \{S_H^{(l)}\}_{H\in\mathcal{H}_l}$ contains all elements in $\mathcal{K}_{l+1}$. By Lemma C.5, for any $S \in \mathcal{K}_{l+1} \cap \mathcal{S}_*$, there exists a unique coordinate $H_S \in [M]$ with $w_{H_S}^{(l)} = (1 \pm 0.2)\hat{f}_*(S)/\alpha_{H_S}^{(l)}$. Combining this identity with (10) gives

$$w_{H_S}^{(l)} z_{H_S}^{(l)} = (1 \pm 0.2)\hat{f}_*(S)\chi_S(\boldsymbol{x}).$$

In other words, $\alpha_{H_S}^{(l+1)} = (1 \pm 0.2)\hat{f}_*(S)$. Note that the same bound and uniqueness result still hold for $S \in \mathcal{S}_* \setminus \mathcal{K}_{l+1}$, but we do not guarantee the existence of H_S. Also, by Lemma C.5, all other coordinates of $\boldsymbol{w}^{(l)}$ and, therefore, the corresponding entries in $\boldsymbol{w}^{(l)} \odot \boldsymbol{z}^{(l)}$, are always 0. In particular, this implies that the number of nonzero entries in $\boldsymbol{w}^{(l)} \odot \boldsymbol{z}^{(l)}$ is at most $|\mathcal{S}_*| \le m$. Hence, `Sel` will never discard needed entries. Recall that $\boldsymbol{x}^{(l+1)}$ is obtained by concatenating $\boldsymbol{x}$ with $\boldsymbol{y}^{(l)}$. Therefore, each of the sets $\{1\}, \dots, \{d\}$ can correspond to at most two entries in $\boldsymbol{x}^{(l+1)}$, with the corresponding α equal either to 1 or to the corresponding Fourier coefficient. □

Our main optimization theorem is a corollary of this lemma.

Proof of Theorem 3.1. First, by Fact C.1 and Lemma C.6, Assumption 3.1 holds for all $l \in [L]$. For the last layer, to achieve the target accuracy ε_*, we need more samples and steps. Let $\tilde{\varepsilon}_* > 0$ be a parameter to be determined later. The proof of Lemma C.5, *mutatis mutandis*, shows that to ensure $w_H^{(L)} = (1 \pm \tilde{\varepsilon}_*)\hat{f}_*(S_H^{(L)})/\alpha_H^{(L)}$ for every H belonging to case (b), it suffices to choose $\varepsilon_1 \vee \varepsilon_2 \le \tilde{\varepsilon}_* \wedge 2^{-2D-4}\kappa^{D+1}$, where $\varepsilon_1, \varepsilon_2$ are defined in the proof of Lemma C.5. In particular, this translates into the constraints:

$$N \gtrsim \frac{M^2 \log(M/\delta_{\mathbb{P}})}{\kappa^{2D+2}\left(\tilde{\varepsilon}_*^2 \wedge \kappa^{2D+2}\right)}, \quad T \gtrsim \frac{\log\left(\tilde{\varepsilon}_*^{-1} \vee 1/\kappa\right)}{\kappa^{2D}\eta}.$$

These two conditions are stronger than those required by earlier layers. For simplicity, we will just replace all earlier requirements with these two. By Parseval's theorem, conditioned on the above event, we have

$$\|f_* - f\|_{L^2} = \left\|\hat{f}_* - \hat{f}\right\|_2 \le \sqrt{|\mathcal{S}_*|}\tilde{\varepsilon}_*.$$

Finally, taking a union bound over all L layers and replacing $\delta_{\mathbb{P}}$ with $\delta_{\mathbb{P}}/L$ and $\tilde{\varepsilon}_*$ with $\varepsilon_*/\sqrt{|\mathcal{S}_*|}$ completes the proof. □

D Compiling the learner network

In this section, we show that we can efficiently compile the learner network from our upper bound (cf. Theorem 3.1 and (5)) into a network whose architecture matches our lower bound (cf. (2) and Assumption 2.1).

First, recall from Appendix C that Assumption 3.1 holds for every $l \in [L]$. In particular, for any nontrivial coordinate $k \in I_l$, we have

$$x_k^{(l)} = \alpha_k^{(l)} \chi_{S_k^{(l)}}(\boldsymbol{x}), \quad \alpha_k^{(l)} \neq 0, \quad S_k^{(l)} \subset [d].$$

Instead of computing $\boldsymbol{x}^{(l)}$ using a layer satisfying Assumption 2.1, we compute a normalized version of it. Specifically, we define

$$\tilde{x}_k^{(l)}(\boldsymbol{x}) = \begin{cases} \chi_{S_k^{(l)}}(\boldsymbol{x}), & k \in I_l, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall \boldsymbol{x} \in \{\pm 1\}^d,\ k \in [d+m].$$

By our analysis in Appendix C and the fact that the first d coordinates of $\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})$ are always $\boldsymbol{x}$, for any $j \in [d+m]$, either $\tilde{x}_j^{(l+1)} \equiv 0$, or there exists $H_j^{(l)} \subset I_l$ such that

$$\tilde{x}_j^{(l+1)} \equiv \chi_{S^{(l)}_{H_j^{(l)}}} \quad \text{where} \quad S^{(l)}_{H_j^{(l)}} := \bigoplus_{i \in H_j^{(l)}} S_i^{(l)}.$$

For each $l \in [L]$, let $\boldsymbol{A}^{(l)} \in \{0,1\}^{(d+m)\times(d+m)}$ be the corresponding incidence matrix:

$$A_{ji}^{(l)} := \mathbb{1}\left\{ i \in H_j^{(l)} \right\}, \tag{23}$$

with the convention that $A_{ji}^{(l)} = 0$ if $j \notin I_{l+1}$. Define the maximum fan-out of the network as

$$\Delta := \max_{l \in [L]} \max_{i \in [d+m]} \left| \left\{ j \in [d+m] \,:\, i \in H_j^{(l)} \right\} \right|. \tag{24}$$

In words, Δ is the maximum number of times a given input coordinate is used to construct the outputs of a layer. The following are the main results of this section.

Lemma D.1 (Compiling the network)**.** *Suppose that the target function $f_* : \{\pm 1\}^d \to \mathbb{R}$ belongs to $\mathrm{HS}_{L,D,m,\kappa}$ with $L \le d$, $D = O(1), m \le \mathrm{poly}(d), \kappa \ge 1/\mathrm{poly}(d)$ and $\|f_*\|_{L^2} \le 1$. Let our learner network $(\boldsymbol{x}^{(l)})_l, f$ be defined by (5) and suppose that it is (successfully) trained to fit f_* to ε_*-accuracy with $\varepsilon_* = o(1)$ using Algorithm 1.*[3] *Let Δ denote the maximum fan-out of the learner network (cf. (24)).*

Then, there exist activations $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{L+1} : \mathbb{R}^{d+m} \to \mathbb{R}^{d+m}$ and weight matrices $\boldsymbol{W}^{(0)} \in \mathbb{R}^{(d+m)\times d}$, $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(L+1)} \in \mathbb{R}^{(d+m)\times(d+m)}$, $\boldsymbol{a} \in \mathbb{R}^{d+m}$ such that the following hold:

- *(a) They satisfy Assumption 2.1(b)-(d) with $C_W = 2 \vee (\pi/2)\sqrt{D\Delta}$, $c_\phi = 1$, $C_\phi = 2$, $m_\phi = 1$.*
- *(b) Define $\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(L+2)} : \{\pm 1\}^d \to \mathbb{R}^{d+m}$ inductively by*

$$\hat{\boldsymbol{x}}^{(1)}(\boldsymbol{x}) = \boldsymbol{\phi}_0\left(\boldsymbol{W}^{(0)}\boldsymbol{x}\right), \quad \hat{\boldsymbol{x}}^{(l+1)}(\boldsymbol{x}) = \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\hat{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right), \quad \forall l \in [L+1].$$

We have $f(\boldsymbol{x}) = \left\langle \boldsymbol{a}, \hat{\boldsymbol{x}}^{(L+2)}(\boldsymbol{x}) \right\rangle$ for all $\boldsymbol{x} \in \{\pm 1\}^d$.

[3] We say the model is successfully trained if all high-probability events used in the proof occur. By (the proof of) Theorem 3.1, this happens with probability at least $1 - \delta_{\mathbb{P}}$.

Moreover, the activation functions and weights can be constructed from the trained network in polynomial time.

Corollary D.2. *Let our learner network $(\boldsymbol{x}^{(l)})_l$, f be defined by (5) and suppose that it is (successfully) trained to fit a deep quadratic function to ε_*-accuracy with $\varepsilon_* = o(1)$ using Algorithm 1. Then, $\Delta \le 5$. As a result, Lemma D.1 holds, and the constructed network satisfies Assumption 2.1(b)-(d) and $C_W, C_\phi = O(1)$.*

To prove Lemma D.1, first we show that each output feature $\tilde{x}_j^{(l+1)}$ can be represented using a single neuron.

Lemma D.3. *Fix $l \in [L]$ and $H \subset I_l$. Define $\boldsymbol{w}_H \in \mathbb{R}^{d+m}$ by $w_{H,i} := (\pi/2)\mathbb{1}\{i \in H\}$, and let $\phi_H(z) := \cos(\pi|H|/2 - z)$. Then, we have*

$$\phi_H\left(\boldsymbol{w}_H \cdot \tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right) = \chi_{S_H^{(l)}}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \{\pm 1\}^d.$$

Proof. For notational simplicity, we drop the superscript l in this proof. Let $K_H(\boldsymbol{x}) := |\{i \in H : \tilde{x}_i(\boldsymbol{x}) = -1\}|$. By construction, we have

$$\langle \boldsymbol{w}_H, \tilde{\boldsymbol{x}}(\boldsymbol{x})\rangle = \sum_{i \in I_l} w_{H,i}\tilde{x}_i(\boldsymbol{x}) = \frac{\pi}{2}\sum_{i \in H}\chi_{S_i}(\boldsymbol{x}) = \frac{\pi}{2}\left(|H| - 2K_H(\boldsymbol{x})\right).$$

Therefore,

$$\phi_H(\boldsymbol{w}_H \cdot \tilde{\boldsymbol{x}}(\boldsymbol{x})) = \cos\left(\frac{\pi}{2}|H| - \frac{\pi}{2}\left(|H| - 2K_H(\boldsymbol{x})\right)\right) = \cos(\pi K_H(\boldsymbol{x})) = \begin{cases} 1, & K_H(\boldsymbol{x}) \text{ is even,} \\ -1, & K_H(\boldsymbol{x}) \text{ is odd.} \end{cases}$$

The quantity $K_H(\boldsymbol{x})$ is even if and only if $\prod_{i \in H}\tilde{x}_i(\boldsymbol{x}) = 1$, which holds if and only if $\chi_{S_H}(\boldsymbol{x}) = 1$. In other words, we have $\phi_H(\boldsymbol{w}_H \cdot \tilde{\boldsymbol{x}}(\boldsymbol{x})) = \chi_{S_H}(\boldsymbol{x})$. □

Lemma D.4. *For any $l \in [L]$, there exist $\boldsymbol{\phi}_l$ and $\boldsymbol{W}^{(l)}$ satisfying Assumption 2.1(b)-(d) with $C_W = (\pi/2)\sqrt{D\Delta}$, $c_\phi = 1$, $C_\phi = 2$, $m_\phi = 0$ such that*

$$\tilde{\boldsymbol{x}}^{(l+1)}(\boldsymbol{x}) := \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right), \quad \forall \boldsymbol{x} \in \{\pm 1\}^d.$$

Proof. For notational simplicity, we write $S_j^{(l)} := S^{(l)}_{H_j^{(l)}}$ in this proof. Recall that for any $j \in [d+m]$, we have

$$\tilde{x}_j^{(l+1)}(\boldsymbol{x}) = \begin{cases} \chi_{S_j^{(l)}}(\boldsymbol{x}), & j \in I_{l+1}, \\ 0, & \text{otherwise,} \end{cases} \quad \forall \boldsymbol{x} \in \{\pm 1\}^d.$$

For $j \notin I_{l+1}$, we simply set $\phi_{l,j} \equiv 0$ and $\boldsymbol{w}_j^{(l)} = 0$. For $j \in I_{l+1}$, we choose

$$\boldsymbol{w}_j^{(l)} = \left(\frac{\pi}{2}\mathbb{1}\left\{i \in H_j^{(l)}\right\}\right)_{i \in [d+m]}, \quad \phi_{l,j}(z) = \cos\left(\frac{\pi}{2}\left|H_j^{(l)}\right| - z\right).$$

Then, by Lemma D.3, we have

$$\tilde{\boldsymbol{x}}^{(l+1)}(\boldsymbol{x}) := \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right), \quad \forall \boldsymbol{x} \in \{\pm 1\}^d.$$

Now, we upper bound the operator norm of $\boldsymbol{W}^{(l)}$. Recall the definition of $\boldsymbol{A}^{(l)} \in \{0,1\}^{(d+m)\times(d+m)}$ from the beginning of this section. Note that we have

$$\left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}} = \frac{\pi}{2}\left\|\boldsymbol{A}^{(l)}\right\|_{\mathrm{OP}} \le \frac{\pi}{2}\sqrt{\left\|\boldsymbol{A}^{(l)}\right\|_1\left\|\boldsymbol{A}^{(l)}\right\|_\infty},$$

where $\|\boldsymbol{A}\|_1 := \max_i \sum_j |A_{ji}|$ and $\|\boldsymbol{A}\|_\infty := \max_j \sum_i |A_{ji}|$. Since $|H_j^{(l)}| \le D$, we have $\left\|\boldsymbol{A}^{(l)}\right\|_\infty \le D$. In addition, by the definition of maximum fan-out (cf. (24)), we have $\left\|\boldsymbol{A}^{(l)}\right\|_1 \le \Delta$. Therefore,

$$\left\|\boldsymbol{W}^{(l)}\right\|_{\mathrm{OP}} \le \frac{\pi}{2}\sqrt{D\Delta}.$$

Finally, we show that $\boldsymbol{\phi}_l$ satisfies Assumption 2.1(c)-(d) with $c_\phi = 1$, $C_\phi = 2$ and $m_\phi = 0$. Since our activation functions are shifted cosines, it is clear that Assumption 2.1(c) holds as long as $C_\phi \ge 1$. In addition, for any $x_0 \in \mathbb{R}$, $z \mapsto \cos(x_0 + z)$ is entire and satisfies $|\cos(a + x_0 + ib)| \le \cosh|b|$ for all $a, b \in \mathbb{R}$. Take $c_\phi = 1$. Then, for any $|b| \le 1/2$, we have $|\cos(a + x_0 + ib)| \le \cosh|b| \le 2$. In other words, Assumption 2.1(d) holds with $c_\phi = 1$, $C_\phi = 2$ and $m_\phi = 0$. □

Now, we are ready to prove the main results of this section.

Proof of Lemma D.1. First, by Lemma D.4, we can choose $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_L$ and $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(L)}$ satisfying the requirements such that

$$\tilde{\boldsymbol{x}}^{(l+1)}(\boldsymbol{x}) := \boldsymbol{\phi}_l\left(\boldsymbol{W}^{(l)}\tilde{\boldsymbol{x}}^{(l)}(\boldsymbol{x})\right), \quad \forall \boldsymbol{x} \in \{\pm 1\}^d,\, l \in [L].$$

Hence, to complete the proof, it suffices to construct the 0th layer and the output layer(s). Recall that the raw input is $\boldsymbol{x} \in \{\pm 1\}^d$ and $\tilde{\boldsymbol{x}}^{(1)}(\boldsymbol{x}) = \boldsymbol{x} \circ \boldsymbol{0} \in \{0, \pm 1\}^{d+m}$. Hence, to ensure $\hat{\boldsymbol{x}}^{(1)} \equiv \tilde{\boldsymbol{x}}^{(1)}$, it suffices to choose $\boldsymbol{\phi}_0 = \mathrm{Id}$ and $\boldsymbol{W}^{(0)} = [\boldsymbol{I}_d\ \boldsymbol{0}]^\top \in \mathbb{R}^{(d+m)\times d}$. Then, define $(\hat{\boldsymbol{x}}^{(l)})_{l\in[L+1]}$ as in Lemma D.1. We have $\hat{\boldsymbol{x}}^{(l)} \equiv \tilde{\boldsymbol{x}}^{(l)}$ for all $l \in [L+1]$. In particular, this implies that for every $S \in \mathcal{S}_*$, there exists some $j_S \in [d+m]$ such that $\hat{x}_{j_S}^{(L+1)} \equiv \chi_S$. In addition, by the proof of Theorem 3.1, we know that after training, the trained model satisfies

$$f = \sum_{S\in\mathcal{S}_*} \tilde{a}_S \chi_S, \quad \text{where } \left|\tilde{a}_S - \hat{f}_*(S)\right| \le \varepsilon_*/\sqrt{|\mathcal{S}_*|},\ \forall S \in \mathcal{S}_*.$$

Thus, in principle, it suffices to set the entry of the output weight $\boldsymbol{a}$ associated with each S equal to the corresponding $\tilde{a}_S$. Note that because of these small errors, the norm of this $\boldsymbol{a}$ can potentially be slightly larger than 1. To fix this, we set $\boldsymbol{W}^{(L+1)} = 2\boldsymbol{I}_{d+m}$ and $\boldsymbol{\phi}_{L+1} = \mathrm{Id}$, and then set the corresponding entry in $\boldsymbol{a}$ to be $\tilde{a}_S/2$. Then, we have $\|\boldsymbol{a}\|_2 = \|\tilde{\boldsymbol{a}}\|/2 \le (1+\varepsilon_*)/2 \le 1$. Also, note that the identity activations satisfy Assumption 2.1 with $c_\phi = 1, C_\phi = 2, m_\phi = 1$.

Finally, to verify that these weights and activations can be constructed efficiently from the original trained network, observe that the incidence matrix (23) and output weights can be constructed by recording the original index of the intermediate features and inspecting the nonzero entries of the learner weights. □

Proof of Corollary D.2. It suffices to show that the maximum fan-out Δ is at most 5. Without loss of generality, we may assume that the target function is the base deep quadratic function.

Recall that $\boldsymbol{x}^{(l)} = \boldsymbol{x} \circ \boldsymbol{y}^{(l-1)}$. In the first layer, the degree-1 feature will be copied once and used to construct the (disjoint) degree-2 features. Hence, the maximum fan-out of the first layer is 2.

Now, fix $l \ge 2$. Recall from our induction hypothesis (Assumption 3.1) that the features stored in $\boldsymbol{y}^{(l)}$ are unique. In addition, note that each input feature can be used at most 5 times. To see this, consider some $P \subset [d]$ in an intermediate representation. Let Q denote its sibling, A, B its two unique children, and $R = P \cup Q$ its (unique) parent. Then, P can only be potentially used to construct the output features P, $A = P \oplus B$, $B = P \oplus A$, $Q = R \oplus P$ and $R = P \oplus Q$. Hence, $\Delta \le 5$.

To complete the proof, recall from Fact 4.2 that the local degree and width of any deep quadratic function are 2 and $d-1$, respectively. □

E Further discussion and examples of hierarchical staircase functions

In this section, we give further examples of hierarchical staircase functions and discuss their relationship to and differences from related function classes.

E.1 Staircase-type functions

Staircase function. Abbe et al. (2021) introduce (basic) staircase functions, which are defined as Boolean functions $f = \sum_{S\subset[d]} \hat{f}(S)\chi_S$ satisfying the property that for every $S \in \mathcal{S}_f$ with $|S| \ge 2$, there exists $S' \subset S$ such that $S' \in \mathcal{S}_f$ and $|S \setminus S'| = 1$. In words, every set S in the Fourier spectrum can be constructed by taking the union of another smaller set S' in the spectrum and one degree-1 monomial. It is clear that such functions are also hierarchical staircase functions with local degree 2 and depth at most the size of the largest set in $\mathcal{S}_f$ plus one.

Merged-staircase function. Later, Abbe et al. (2022) generalize staircase functions to merged-staircase functions by replacing S' with a union of earlier sets. Formally, a Boolean function f is said to be a merged-staircase function if there is an ordering $\{S_1, \dots, S_r\}$ of $\mathcal{S}_f$ such that

$$\left|S_i \setminus \bigcup_{j=1}^{i-1} S_j\right| \le 1, \quad \forall i \in [r], \tag{25}$$

and the optimization guarantees in Abbe et al. (2022) further require f to be $O(1)$-sparse, i.e., $\left|\bigcup_{S\in\mathcal{S}_f} S\right| = O(1)$. The function classes and results in this paper and Abbe et al. (2022) are not directly comparable. An obvious difference is that we allow the function to have genuinely high-degree terms, while the sparsity condition in Abbe et al. (2022) effectively allows only constant-degree terms. Hence, in principle, all k-sparse merged-staircase functions are depth-2 hierarchical staircase functions with local degree k. Even if we drop the sparsity constraint, these two function classes are still not comparable, since $\bigcup_{j=1}^{i-1} S_j$ can potentially involve a superconstant number of earlier sets. For example, take $S_i = \{i\}$ for $i \in [d/2]$ and $S_{d/2+1} = \{1, \dots, d/2\}$. The collection $\{S_1, \dots, S_{d/2+1}\}$ satisfies (25), but it is not a hierarchical staircase function with constant local degree. However, if we restrict the number of $\{S_j\}_{j<i}$ that can be used by S_i to be a constant D and replace the union and set difference with symmetric difference in (25), then the resulting functions will be hierarchical staircase functions with local degree $D+1$. In addition, we believe that some constraint on sparsity and/or on the number of usable sets is necessary for efficient learning. To see this, note that $S_i = \{i\}$ for $i \in [d]$ and $S_{d+1} = \{1, \dots, d/2\}$ also satisfy (25), and learning S_{d+1} after learning $(S_i)_{i\in[d]}$ is essentially learning $(d/2)$-parity, which is known to be hard for statistical query models (Kearns (1998)) and certain gradient-based methods (Shalev-Shwartz et al. (2017)).

Leap complexity. Abbe et al. (2023) further generalize the concept of merged-staircase functions by allowing the right-hand side of (25) to be a constant larger than 1. The smallest possible such constant is called the leap complexity. As in the preceding discussion, if we restrict the number of usable sets on the left-hand side of (25) to be constant and replace the union and set difference with symmetric difference, then functions with constant leap complexity are hierarchical staircase functions whose local degree is at most the number of usable sets plus the leap complexity.

E.2 Globally read-once decision trees

A **Boolean decision tree** f_T over $\{\pm1\}^d$ is a rooted binary tree whose leaves are labeled by elements of $\{\pm1\}$ and whose internal nodes v are labeled by coordinates $i_v \in [d]$; the two outgoing edges of each internal node are labeled by the elements of $\{\pm1\}$. To evaluate $f_T(\boldsymbol{x})$ for $\boldsymbol{x} \in \{\pm1\}^d$, we start at the root. At an internal node v, we query x_{i_v} and follow the edge labeled by its value. The output is the value of the leaf we eventually reach. We say a decision tree is **globally read-once** if each coordinate of $\boldsymbol{x}$ can appear at most once in the decision tree. For more background on decision trees, one may refer to Section 3.2 of O'Donnell (2014).

Lemma E.1. *Let f_T be a depth-h globally read-once Boolean decision tree, where $h \ge 1$. Then, $f_T \in \mathrm{HS}_{h,2,m,\kappa}$ with $m \le 4^h$ and $\kappa \ge 2^{-h}$.*

Proof. We first verify the hierarchical staircase property by induction on h. Let i be the coordinate queried at the root, and let f_- and f_+ be the functions computed by the subtrees reached when $x_i = -1$ and $x_i = +1$, respectively. Then

$$f_T(\boldsymbol{x}) = \frac{1-x_i}{2} f_-(\boldsymbol{x}) + \frac{1+x_i}{2} f_+(\boldsymbol{x}). \tag{26}$$

Let V_- and V_+ denote the sets of coordinates queried in the two subtrees. Since T is globally read-once, V_- and V_+ are disjoint and neither contains i. Note that for any Boolean function f and $S \subset [d]$, $\hat{f}(S) = 0$ if there exists some $i \in S$ on which f does not depend. Hence, it follows from (26) that, for each $a \in \{\pm1\}$ and each nonempty $S \in \mathcal{S}_{f_a}$,

$$i \notin S, \qquad \hat{f}_T(S) = \frac{1}{2}\hat{f}_a(S), \qquad \hat{f}_T(S\cup\{i\}) = \frac{a}{2}\hat{f}_a(S).$$

In particular, both S and $S\cup\{i\}$ belong to $\mathcal{S}_{f_T}$. Moreover, apart from $\varnothing$ and $\{i\}$, every set in $\mathcal{S}_{f_T}$ is of one of these two forms. In particular, in the second case, $S\cup\{i\} = S \oplus \{i\}$ can be constructed by combining an existing feature and a degree-1 term.

If $h = 1$, both subtrees are constant, so $\mathcal{S}_{f_T} \subseteq \{\varnothing, \{i\}\} \subseteq \mathcal{K}_1$. Now suppose $h \geq 2$. By the induction hypothesis, every nonempty set in the spectrum of either subtree can be constructed within $h - 1$ rounds. The same construction remains valid for f_T, since all of its nonempty intermediate sets also belong to $\mathcal{S}_{f_T}$ by the preceding display. Thus, every such S belongs to $\mathcal{K}_{h-1}$ for f_T. Since $\{i\} \in \mathcal{K}_1 \subseteq \mathcal{K}_{h-1}$ and $S \cup \{i\} = S \oplus \{i\}$, every set of the second form belongs to $\mathcal{K}_h$. We conclude that $\mathcal{S}_{f_T} \subseteq \mathcal{K}_h$, so f_T has depth at most h and local degree at most 2.

It remains to bound the width and signal strength. For each leaf l, let $b_l \in \{\pm 1\}$ be its label, let P_l be the set of coordinates queried on the root-to-l path, and let $a_{l,j} \in \{\pm 1\}$ be the edge label followed when coordinate $j \in P_l$ is queried. Since the leaf events partition the hypercube,

$$f_T(\boldsymbol{x}) = \sum_l b_l \prod_{j \in P_l} \frac{1 + a_{l,j} x_j}{2}.$$

A leaf at depth t contributes at most $2^t \leq 2^h$ Fourier monomials, and a depth-h binary tree has at most 2^h leaves. Hence,

$$|\mathcal{S}_{f_T}| \leq \sum_l 2^{|P_l|} \leq 4^h.$$

Finally, after expanding the preceding leaf representation, every Fourier coefficient is a sum of terms of the form $\pm 2^{-|P_l|}$. Since $|P_l| \leq h$, every Fourier coefficient is an integer multiple of 2^{-h}. Therefore, every nonzero Fourier coefficient has magnitude at least 2^{-h}, which proves the claimed lower bound on κ. □

E.3 Addressing functions

Fix $k \in \mathbb{N}_+$. Let $\boldsymbol{a} = (a_1, \ldots, a_k) \in \{\pm 1\}^k$ be the address variables and let $\boldsymbol{y} = (y_b)_{b \in \{\pm 1\}^k} \in \{\pm 1\}^{2^k}$ be the data variables. The k-bit **addressing function** is

$$\mathrm{Addr}_k(\boldsymbol{a}, \boldsymbol{y}) := y_{\boldsymbol{a}} = \sum_{b \in \{\pm 1\}^k} y_b \prod_{j=1}^{k} \frac{1 + b_j a_j}{2}. \tag{27}$$

In words, Addr_k returns the data bit indexed by the address $\boldsymbol{a}$.

Lemma E.2. $\mathrm{Addr}_k \in \mathrm{HS}_{k+1, 2, 4^k, 2^{-k}}$.

Proof. Expanding (27) gives

$$\mathrm{Addr}_k(\boldsymbol{a}, \boldsymbol{y}) = 2^{-k} \sum_{b \in \{\pm 1\}^k} \sum_{A \subseteq [k]} \left(\prod_{j \in A} b_j \right) y_b \prod_{j \in A} a_j. \tag{28}$$

The 4^k monomials in (28) are distinct, and every coefficient has magnitude 2^{-k}. Hence, Addr_k has width 4^k and signal strength 2^{-k}.

For $A \subseteq [k]$ and $b \in \{\pm 1\}^k$, let $S_{A,b}$ denote the support of the monomial $y_b \prod_{j \in A} a_j$. We show by induction on $|A|$ that $S_{A,b} \in \mathcal{K}_{|A|+1}$. The base case follows because $S_{\varnothing,b} = \{y_b\} \in \mathcal{K}_1$. If A is nonempty, fix any $j \in A$. Then

$$S_{A,b} = S_{A \setminus \{j\}, b} \oplus \{a_j\}.$$

The first set belongs to $\mathcal{K}_{|A|}$ by the induction hypothesis, the singleton $\{a_j\}$ belongs to $\mathcal{K}_1 \subseteq \mathcal{K}_{|A|}$, and $S_{A,b}$ belongs to the spectrum by (28). Therefore, $S_{A,b} \in \mathcal{K}_{|A|+1}$. Since $|A| \leq k$, the full spectrum is contained in $\mathcal{K}_{k+1}$, proving the hierarchical staircase claim. □

F Experimental details

Target function and data. We choose the input dimension to be $d = 256 = 2^8$. The target function f_* is the base deep quadratic function on $\{\pm 1\}^d$. Since our learner network and training algorithm are equivariant under permutations of the form in Definition 4.1, being able to learn the base deep quadratic function implies being able to learn all deep quadratic functions (of the same depth).

Learner model. The learner architecture is a fully connected ReLU network with residual connections from each layer to the final output. More formally, it is defined as

$$\boldsymbol{h}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad \boldsymbol{h}_l(\boldsymbol{x}) = \boldsymbol{P}_l \mathrm{ReLU}(\boldsymbol{W}_l \boldsymbol{h}_{l-1}(\boldsymbol{x}) + \boldsymbol{b}_l), \quad f(\boldsymbol{x}) = \frac{1}{\sqrt{Lm}} \sum_{l=1}^{L} \langle \mathbf{1}, \boldsymbol{h}_l(\boldsymbol{x}) \rangle ,$$

where L and m are the depth and width, respectively, and $(\boldsymbol{P}_l)_{l\in[L]} \subset \mathbb{R}^{m\times m}, (\boldsymbol{b}_l)_{l\in[L]} \subset \mathbb{R}^m$, $\boldsymbol{W}_1 \in \mathbb{R}^{m\times d}, (\boldsymbol{W}_l)_{l=2}^{L} \subset \mathbb{R}^{m\times m}$ are the trainable parameters. In the experiments, we use $L \in \{2, 4, 6, 8\}$ and the width m is chosen so that $Lm^2 \approx 8 \cdot 512^2$. W_l are initialized using Kaiming initialization, $\boldsymbol{P}_l$ are initialized using random orthogonal matrices, and $\boldsymbol{b}_l$ are initialized to 0.

Training algorithm. We train the learner model using the Adam optimizer (Kingma & Ba, 2017). In each step, we sample a fresh batch of size 4096. The step size is 10^{-5} for the first 3×10^6 steps and 3×10^{-6} after that. All models are trained for 2.5×10^7 steps. We do not extensively tune the hyperparameters and the current hyperparameters are chosen so that the training of the largest model is reasonably stable. It took us roughly one week to complete all 10 runs on a server with 10 A6000 GPUs.

Further explanation of Figure 1. Figure 1a reports the smoothed training loss, averaged across 10 independent runs. More specifically, we train 10 independent models. For each run, we gather the training loss and smooth it using an exponential moving average with a half-life of 1000 steps. Then, we average over all runs. The curves are the averaged smoothed losses, and the shaded areas report the standard deviation.

In Figure 1b, we freeze the trained network and fit a separate affine ridge probe from each intermediate $\boldsymbol{h}_l$ to each target-level component, where each target component is the part of the deep quadratic function with the corresponding degree. Figure 1b reports the R^2 heatmap of a single run. The results of the other runs are reported in Figure 2.

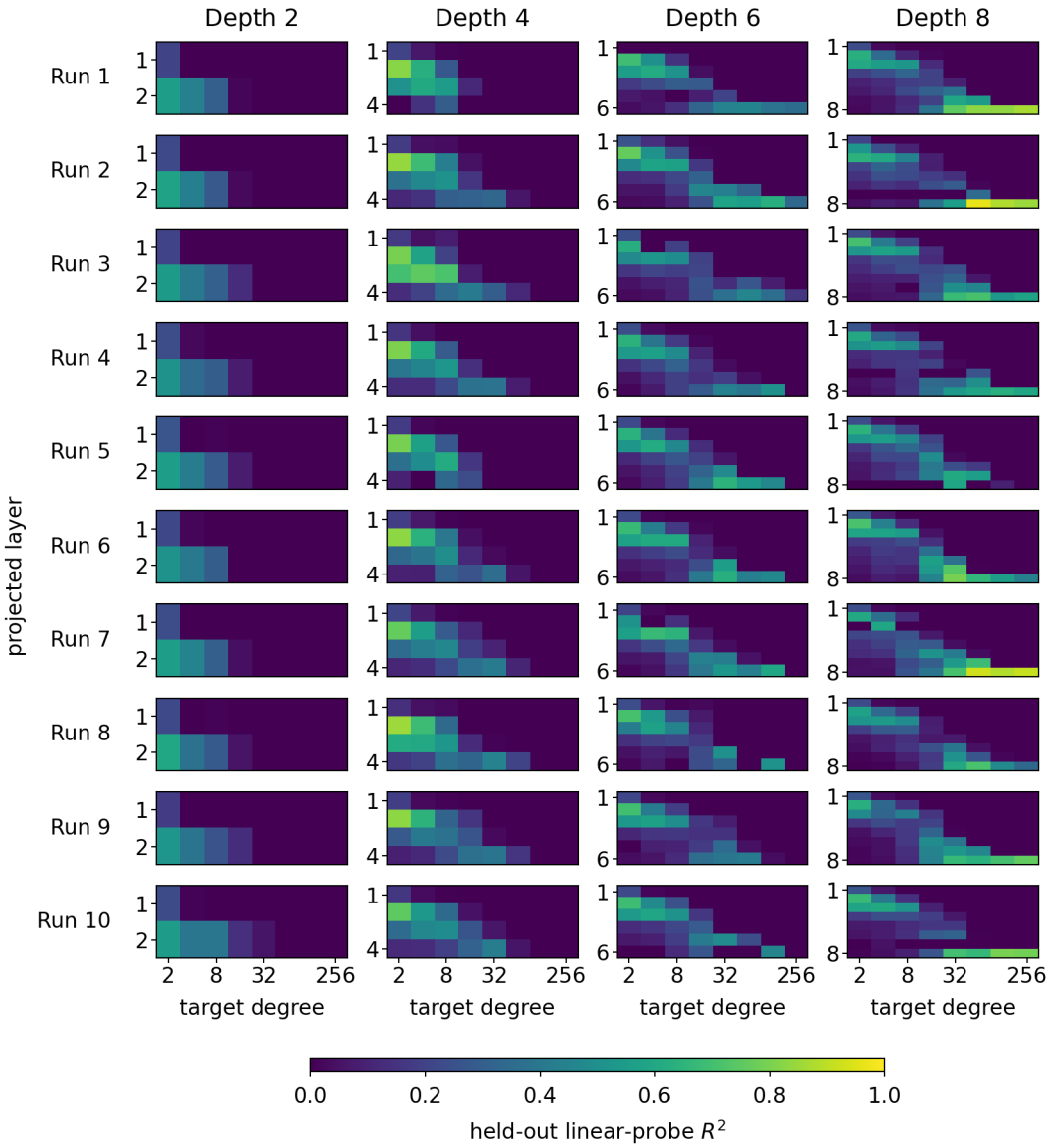


Figure 2: Layer degree probes (all runs)