Learning Sparse Compositional Functions with Norm-Constrained Neural Networks

Shuo Huang
Istituto Italiano di Tecnologia
shuo.huang@iit.it

Lorenzo Fiorito
University of Genova
lorenzo.fiorito@edu.unige.it

Lorenzo Rosasco
University of Genova, Italy
Istituto Italiano di Tecnologia, Italy
Lorenzo.Rosasco@unige.it

Tomaso Poggio
Massachusetts Institute of Technology
tp@csail.mit.edu

Abstract

The ability of deep neural networks to learn hierarchical features is widely regarded as a key mechanism underlying their success in high-dimensional learning. Existing theory partially supports this view by establishing approximation rates based on parameter counts and sample complexity guarantees for compositional models without incurring the curse of dimensionality (CoD). To study overparameterized regimes, where the number of parameters exceeds the sample size, we develop a framework that measures complexity via the parameter norm. Within this approach, we establish approximation rates and excess risk bounds for learning sparse compositional functions whose compositional structure is represented by directed acyclic graphs (DAGs), using Frobenius norm-constrained deep neural networks. Our results have broad applicability since every function that is efficiently Turing computable admits sparse compositional representations. In particular, we cover a range of representative models, including multi-index models, binary tree structures, and general compositional architectures. The rates we derive show that deep networks can exploit the compositional structure of the target functions, effectively avoiding the CoD through hierarchical representations.

1 Introduction

The success of deep learning in high-dimensional tasks such as vision and language has motivated the search for principles underlying its effectiveness. A central insight is that depth enables efficient hierarchical representations, progressively extracting features of increasing complexity [81, 19, 71]. This organization parallels hierarchical processing in the visual cortex [83, 31, 36, 28] and arises in structured data such as syntactic parse trees and object–part decompositions [22, 17]. These observations have led to hierarchical compositional data models that play a key role in theoretical analyses of deep learning efficiency [4, 1, 73]. A prominent example is the class of sparse compositional functions, where each component depends on only a few variables [58, 68, 67]. This perspective is consistent with architectures incorporating locality, such as convolutional neural networks and transformers [51, 85]. Furthermore, the class of functions that are compositionally sparse is broad, since functions that are efficiently computable by a Turing machine admit compositionally sparse representations [66, 26]. From a statistical learning perspective, the strength of this principle lies in its capacity to reduce sample complexity, that is the number of samples required to achieve a target accuracy, therefore improving learning efficiency [84].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Classical results establish that sample complexity scales exponentially with the input dimension, the so-called curse of dimensionality (CoD) [29]. For compositional models, however, deep neural network results show that the required sample size depends on the dimensions of the component functions rather than the ambient dimension, thereby alleviating the CoD [68, 72, 49, 61]. These guarantees, however, are expressed in terms of the network parameter count, which is required to be smaller than the sample size. Thus these results do not extend to modern overparameterized regimes, where number of parameters vastly exceed the number of training samples, as in large foundation models including GPT-4 [2], Gemini [80], and LLaMA [82]. This discrepancy raises two fundamental questions: what complexity measure can appropriately characterize large-scale, overparameterized neural networks beyond parameter count? And how can one establish corresponding sample complexity guarantees for hierarchical learning in this regime?

Recent advances in optimization theory provide insight into the first question. In overparameterized regimes, empirical risk minimization admits infinitely many interpolating solutions, yet gradient-based methods exhibit an implicit bias toward low-complexity ones characterized by certain norms [77, 40, 63, 37]. These findings indicate that effective complexity is governed by some norm structures across layers rather than raw parameter counts.

Based on these observations, we investigate the sparse compositionality principle by analyzing the sample complexity of learning sparse compositional functions with norm-constrained neural networks. Our main contributions are summarized as follows.

- We establish approximation rates for Hölder spaces characterized by Frobenius norm constraints, rather than by the parameter count typically used in classical approximation theory. This shifts the complexity measure from network size to norm-controlled capacity, thereby extending the analysis to the overparameterized regime.
- Building on the DAG-based formulation of sparse compositional functions, we further provide the first approximation rates for such models in terms of the norm of parameters of neural networks. Explicit rates for multi-index models, binary trees, and more general architectures demonstrate the mitigation of the curse of dimensionality.
- Moreover, we show that the sample complexity of learning sparse compositional functions depends exponentially on the intrinsic input dimension of the local constituent functions, rather than on the ambient dimension. This establishes that hierarchical locality mitigates the curse of dimensionality even when networks are highly overparameterized, but controlled through norm constraints.

The remainder of the paper is organized as follows. Section 2 reviews related work. In Section 3, we establish norm-based approximation rates for both single Hölder spaces and sparse compositional Hölder spaces. Section 4 presents the statistical analysis and derives sample complexity bounds, with Subsection 4.1 discussing broader perspectives and outlining potential research directions. Finally, Section 5 concludes the paper.

2 Related work

Compositionality principle Hierarchical compositional structure has been supported from multiple theoretical perspectives as a fundamental principle underlying the effectiveness of deep neural networks. In approximation theory, deep compositional networks, convolutional neural networks, and transformers have been shown to adapt to compositional targets, achieving rates governed by the intrinsic dimensionality of intermediate functions rather than the ambient input dimension [58, 56, 23, 75]. Tree tensor networks provide another hierarchical representation of high-dimensional functions. Bachmayr et al. [8] establish approximation results for Sobolev and compositional classes, and Ali and Nouy [3] study approximation classes of tensorized multivariate functions. Their bounds measure complexity through tensor ranks and representation size, rather than the product Frobenius norm of network parameters considered here A similar phenomenon appears in statistical learning theory. Sample complexity bounds for compositional models scale with the dimensions of constituent functions, thereby mitigating the curse of dimensionality [72, 49, 41]. Beyond those, this structure also yields computational advantages. Recent works show that gradient-based methods can efficiently learn structured models such as single- and multi-index models as well as staircase-type functions, with complexity characterized through quantities such as the information exponent and the leap

exponent [18, 1, 25, 91, 27]. Furthermore, Poggio and Fraser [67], Danhofer et al. [26] claim that sparse compositionality is implied by efficient Turing computability.

Complexity measures for neural networks The complexity of neural networks has been characterized through various measures aimed at controlling the generalization error, i.e., the discrepancy between population and empirical risk. Classical analyses primarily relied on parameter counting arguments and VC-dimension bounds [14, 5, 88], which are often inadequate for modern deep networks [92, 70]. Subsequent work introduced norm-, margin-, and sharpness-based measures to better capture capacity, see, e.g., [63, 6, 46].

Theoretical analysis of overparameterization The theoretical analysis of overparameterized neural networks has been extensively studied from optimization and generalization perspectives. In the interpolation regime, gradient-based methods exhibit an implicit bias toward low-complexity solutions rather than arbitrary interpolants [77, 37, 43]. This phenomenon is closely related to benign overfitting and double descent behavior observed in modern deep models [16, 12, 40]. Another influential line of work analyzes the infinite-width limit via the neural tangent kernel (NTK) and mean-field perspectives, characterizing training dynamics in highly overparameterized regimes [42, 21, 57]. However, comparatively less is understood from an approximation and statistical efficiency viewpoints in overparameterized regimes.

3 Approximation analysis of Frobenius norm-constrained NNs

In this section, we introduce the class of neural networks under consideration and establish their approximation properties, including explicit rates for approximating the Hölder spaces and sparse compositional Hölder spaces.

3.1 Neural networks with Frobenius norm constraint

Let $D \in \mathbb{N}$ denote the depth of the neural network, and let $\{N_\ell\}_{\ell=0}^{D+1}$ denote the layer widths, where in particular $N_0 = d$ corresponds to the input dimension and $N_{D+1} = k$ to the output dimension. Consider functions $\phi : \mathbb{R}^d \to \mathbb{R}^k$ that can be parameterized by a deep ReLU neural network of the form

$$\phi(x) = A_D\sigma\big(A_{D-1}\sigma(\cdots\sigma(A_0x + b_0)\cdots) + b_{D-1}\big). \tag{1}$$

Here, the input $x \in \mathbb{R}^d$, linear matrix $A_\ell \in \mathbb{R}^{N_{\ell+1}\times N_\ell}$, bias $b_\ell \in \mathbb{R}^{N_{\ell+1}}$, and $\sigma(x) = \max\{0, x\}$ is the ReLU activation function applied componentwise. Denote $\theta = (A_\ell, b_\ell)_{\ell=0}^{D}$ and let f_θ be the corresponding realized function. The function class generated by neural networks of the form (1) with depth D and maximal width $W := \max_\ell N_\ell$ is denoted by $\mathcal{NN}(W, D)$. The Frobenius norm of a matrix $A = (a_{i,j}) \in \mathbb{R}^{m\times n}$ is given by $\|A\|_F := \left(\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2\right)^{1/2}$. We equip the class $\mathcal{NN}(W, D)$ with the multiplicative Frobenius norm constraint

$$\kappa(\theta) := \|A_D\|_F \prod_{\ell=0}^{D-1} \|\tilde{A}_\ell\|_F = \|A_D\|_F \prod_{\ell=0}^{D-1} \sqrt{\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1}, \quad \widetilde{A}_\ell = \begin{pmatrix} A_\ell & b_\ell \\ 0 & 1 \end{pmatrix}. \tag{2}$$

The corresponding norm-constrained function class is

$$\mathcal{NN}(W, D, K) := \{f_\theta \in \mathcal{NN}(W, D) \ : \ \kappa(\theta) \le K\}. \tag{3}$$

The term $+1$ in $\kappa(\theta)$ in (2) is included to handle bias terms in (1) through the standard homogeneous-coordinate representation. Indeed, by appending a constant coordinate as given in $\widetilde{A}_\ell$, a network with bias in (1) can be written as a bias-free augmented network on an input $(x^\top, 1)^\top$. Hence $\kappa(\theta)$ is the Frobenius product norm of the augmented network. This definition allows us to handle biases consistently within norm-based capacity estimates for networks without bias [63, 11, 35]; See Appendix C for more details. We impose a multiplicative norm constraint rather than an additive one. This choice reflects the compositional structure of deep networks: forward propagation and the resulting Lipschitz constant scale multiplicatively across layers. Moreover, ReLU networks without bias are positively homogeneous and invariant under layer-wise rescaling, a symmetry preserved by product norms but not by additive constraints. Generalization bounds based on Rademacher or

margin analysis naturally involve such products [63, 11, 35]. We adopt the Frobenius norm at each layer, which measures the overall magnitude of parameters. Compared to spectral or path norms, it provides a simple and widely used measure corresponding to ℓ_2 regularization (often corresponding to weight decay) in practice [52, 93].

3.2 Approximation rates for Hölder spaces

Let $\alpha = r + \beta$ with $r \in \mathbb{N}_0$ and $\beta \in (0, 1]$. The unit ball of Hölder space $\mathcal{C}^\alpha([-1, 1]^d)$ of order α consists of all functions $h : [-1, 1]^d \to \mathbb{R}$ that are r times continuously differentiable and satisfy

$$\|h\|_{\mathcal{C}^\alpha} := \sum_{|\gamma| \le r} \|D^\gamma h\|_{L^\infty([0,1]^d)} + \max_{|\gamma|=r} \sup_{x \neq y} \frac{|D^\gamma h(x) - D^\gamma h(y)|}{\|x - y\|_\infty^\beta} \le 1.$$

Here $\gamma = (\gamma_1, \ldots, \gamma_d) \in \mathbb{N}_0^d$ is a multi-index with $|\gamma| = \sum_{i=1}^d \gamma_i$, and $D^\gamma h = \partial_1^{\gamma_1} \cdots \partial_d^{\gamma_d} h$. This unit-ball assumption is adopted for convenience and can be extended directly to any bounded ball.

The following theorem establishes the approximation rate of neural networks with Frobenius norm constraints for Hölder spaces, whose proof is presented in Appendix B.1. We write $A \lesssim B$ (resp. $A \gtrsim B$) if $A \le cB$ (resp. $A \ge cB$) for some $c > 0$, and $A \asymp B$ if both hold.

Theorem 1 (Approximation of Hölder spaces)**.** *For any $K \ge 1$ and $h \in \mathcal{C}^\alpha([0, 1]^d)$ with $\alpha = r + \beta$, where $r \in \mathbb{N}_0$ and $\beta \in (0, 1]$, there exists a neural network $\phi \in \mathcal{NN}(W, D, K)$ with depth $D = 2\lceil \log_2(d + r) \rceil + 2$ and width $W \gtrsim K^{\frac{2d+\alpha}{d(D+1)+2}}$ such that*

$$\|h - \phi\|_{L^\infty([0,1]^d)} \lesssim K^{-\frac{2\alpha}{2+d(D+1)}}.$$

The theorem provides a parameter norm-based analogue of classical approximation rates. The resulting rate is polynomial in K, improves with the Hölder smoothness α, and deteriorates with the input dimension d. Although the exponent also depends on the depth D, it is relatively small and grows logarithmically with d. It holds for neural networks with any width W exceeds a polynomial threshold in K, thereby covering the model in the overparameterized regime. In other words, once the width exceeds the stated polynomial threshold in K, the approximation rate is governed by the norm budget rather than by the width. A key step in the proof is to use shallow but wide architectures to approximate square functions, which helps to obtain norm constraints independent of the width. This type of width-independent norm control is not provided by classical approximation results formulated in terms of parameter count, such as [90, 74, 55]. See the discussion following Lemma 1 in Appendix A for quantitative details. We also refer to [87] for a comparison between wider and deeper architectures from approximation and generalization viewpoints.

The closest norm-based result is given in [44], which studies approximation under operator-norm constraints. In contrast, we work on Frobenius norms, which align more naturally with ℓ_2 regularization in practice. Both proofs proceed through square functions, multiplication, monomials, and local Taylor polynomials. Our contribution requires new bounds for the accumulation of Frobenius norms in these constructions and extends the analysis to sparse compositional classes on DAGs. Another related line of work studies approximation rates characterized explicitly by function-space norms in the infinite-width limit, where width no longer affects the rate. In this regime, integral representations of shallow neural networks have been well studied when endowed with total variation or Barron-type norms [7, 89, 76]. We highlight this line of work because it raises intriguing questions about the functional-analytic characterization of deep neural networks and its implications for norm-based approximation theory.

3.3 Approximation rates for sparse compositional Hölder spaces

In this section, we formalize the hierarchical compositional model as a class of sparse compositional functions using the notation of DAGs, and establish the approximation rates of neural networks under Frobenius norm constraints.

3.3.1 Sparse compositional function

In hierarchical compositional functions, the dependency structure across levels can be highly heterogeneous: constituent functions may differ in their input dimensionality, interaction patterns,

and structural complexity. Such variability makes a uniform layered description inadequate. Inspired by prior work [58, 68], DAGs provide a flexible formalism to encode these non-uniform dependencies and varying local dimensions. Let $(\mathcal{V}, \mathcal{E})$ be a DAG with nodes $\mathcal{V} = \bigcup_{\ell=0}^{L} \mathcal{V}^{(\ell)}$, where $\mathcal{V}^{(0)} = \{x_1, \ldots, x_d\}$ and $\mathcal{V}^{(L)} = \{v_{\text{out}}^{(L)}\}$. Let $|\mathcal{V}^{(\ell)}|$ denote the number of nodes at level ℓ. The edge set satisfies $\mathcal{E} = \bigcup_{\ell=1}^{L} \mathcal{E}^{(\ell)}$ with $\mathcal{E}^{(\ell)} \subseteq \mathcal{V}^{(\ell-1)} \times \mathcal{V}^{(\ell)}$. For $v^{(\ell)} \in \mathcal{V}^{(\ell)}$, define $\text{pa}(v^{(\ell)}) = \{u \in \mathcal{V}^{(\ell-1)} : (u, v^{(\ell)}) \in \mathcal{E}^{(\ell)}\}$. We formalize a sparse compositional function $f : \mathbb{R}^d \to \mathbb{R}$ associated with a DAG as

$$f(x) = g_{\mathcal{V}^{(L)}} \circ g_{\mathcal{V}^{(L-1)}} \circ \cdots \circ g_{\mathcal{V}^{(0)}}(x). \tag{4}$$

With a slight abuse of notation, we write the cardinality of the set $\text{pa}(v^{(\ell)})$ as $d_{\text{in}(v^{(\ell)})}$ to represent the input dimension for the associated local function $g_{v^{(\ell)}} : \mathbb{R}^{d_{\text{in}(v^{(\ell)})}} \to \mathbb{R}$. Note that the cardinality $|\mathcal{E}^{(\ell)}| = \sum_i d_{\text{in}(v_i^{(\ell)})}$. Hence, $g_{\mathcal{V}^{(\ell)}} : \mathbb{R}^{|\mathcal{E}^{(\ell)}|} \to \mathbb{R}^{|\mathcal{V}^{(\ell)}|}$ with $g_{\mathcal{V}^{(\ell)}} = (g_{v_1^{(\ell)}}, \ldots, g_{v_{|\mathcal{V}^{(\ell)}|}^{(\ell)}})$ is a collection of functions associated with ℓ-th level nodes. A simple example of DAG illustration with three levels is given in the left of Figure 1.

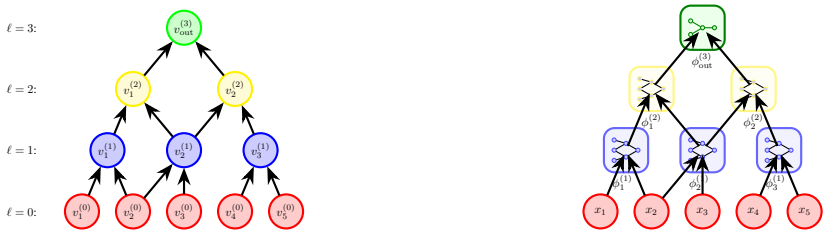

Figure 1: Illustration of a compositional function defined on a DAG (left) and its realization via block-structured neural networks (right), where each node corresponds to a local component.

We note that each node depends only on a subset of nodes from the previous level, rather than all of them. This sparsity induces a low-dimensional input for each constituent function. The compositional structure is sufficiently general to encompass many models studied in the literature. A fundamental example is the multi-index model $f(x) = g(Ax)$, which underlies classical dimension reduction [54, 33, 47, 20], neural network adaptivity to low-dimensional structure [7, 64], and feature learning by gradient-based methods [18, 24, 69]. Another canonical instance is the binary tree compositional structure [68, 58], widely used to demonstrate the expressive and statistical advantages of deep networks over shallow ones. Staircase functions [1], constructed via iterative compositions that progressively reveal information across layers, also fit naturally within the DAG framework and have been employed to characterize the computational benefits of depth through the notions of information and leap exponents. Moreover, Zhang et al. [94] employ a DAG framework to formalize the mathematical structure of reasoning in large language models.

3.3.2 Approximation rates for compositional functions under norm-constrained NNs

Next, we present the approximation rates for sparse compositional functions under the following assumption on each node function.

Assumption 1. *We assume the input $x \in [-1, 1]^d$. Moreover, for $\ell = 1, \ldots, L$, each node function $g_{v^{(\ell)}} \in C^{\alpha_{v^{(\ell)}}}\left([-1, 1]^{d_{in(v^{(\ell)})}}\right)$ with $\alpha_{v^{(\ell)}} = r_{v^{(\ell)}} + \beta_{v^{(\ell)}}$, where $r_{v^{(\ell)}} \in \mathbb{N}_0$ and $\beta_{v^{(\ell)}} \in (0, 1]$.*

The above conditions are standard in approximation theory. The restriction $x \in [-1, 1]^d$ is without loss of generality, as any bounded domain can be rescaled accordingly. The regularity and boundedness assumptions are imposed at the level of individual node functions and are loose since only requires $\alpha_{v^{(\ell)}} > 0$, whose input dimensions $d_{\text{in}(v^{(\ell)})}$ may vary across the graph. In fact, The assumption that each node function is valued in the unit ball is made only for convenience and can be relaxed to any bounded range; see Appendix B.2 for the rescaling argument. The following theorem quantifies the approximation rate of sparse compositional functions in terms of node-level smoothness and dimensionality. The proof appears in Appendix B.2.

Theorem 2. *Let $f : [-1, 1]^d \to \mathbb{R}$ be a sparse compositional function associated with a DAG $(\mathcal{V}, \mathcal{E})$ with L levels as defined in* (4)*. Under Assumption 1, for any $K \geq 1$, there exists a neural network*

$\phi \in \mathcal{NN}(W, D, K)$ *with depth* $D = 2L \max_{\ell, v^{(\ell)}} \lceil \log_2(d_{in(v^{(\ell)})} + r_{v^{(\ell)}}) \rceil + 2L$, *such that, for any width* $W \geq \max_{\ell, v^{(\ell)}} K^{\frac{2d_{in(v^{(\ell)})} + \alpha_{v^{(\ell)}}}{(D+L)\, d_{in(v^{(\ell)})} + 2L}}$, *the approximation error satisfies*

$$\|f - \phi\|_\infty \lesssim \max_{v^{(\ell)}: \pi_{1\to L}} K^{-\frac{2\alpha^*_{v^{(\ell)}}}{2L + (D+L)\, d_{in(v^{(\ell)})}}},$$

where $\pi_{1\to L}$ *denotes the critical path selected by the error propagation in* (17) *from level* 1 *to level* L, *and* $\alpha^*_{v^{(\ell)}} := \alpha_{v^{(\ell)}} \prod_{m=\ell+1}^{L} \min\{\alpha_{v^{(m)}}, 1\}$ *is the effective regularity index along this path.*

Some remarks are in order regarding the theorem.

Network construction The depth D grows only logarithmically with the local structural quantities, up to the factor L caused by the compositional structure. Hence, the construction uses relatively shallow but sufficiently wide subnetworks. This is important under a product norm constraint, since the approximation exponent deteriorates with depth. The logarithmic-depth construction therefore keeps the depth-dependent cost mild while allowing width to absorb the required approximation complexity. The width W can be taken arbitrarily large, so the result naturally covers the overparameterized regime. The construction crucially exploits the composition and concatenation properties of norm-constrained networks (Proposition 2 in Appendix A), allowing us to approximate node functions across different levels of the DAG with subnetworks (see the right panel of Figure 1).

Approximation rate The approximation rate is characterized explicitly in terms of the norm constraint K, rather than the parameter count. The path-wise maximum can be further upper-bounded by the worst case over all possible paths. The exponent depends on the singularity index $\alpha^*_{v^{(\ell)}}$ and the local input dimension $d_{\text{in}(v)}$ of the most restrictive node function, instead of the global smoothness α^* and ambient dimension d. The index $\alpha^*_{v^{(\ell)}}$ reflects the iterative accumulation inherent in Hölder compositions, a phenomenon also observed in [45, 72]. If the node functions are at least Lipschitz continuous, i.e., $\alpha_{v^{(\ell)}} \geq 1$, then the singularity index satisfies $\alpha^*_{v^{(\ell)}} = \alpha_{v^{(\ell)}}$, so the smoothness is inherited directly from each node rather than arising through iterative accumulation. Compared with Theorem 1, the replacement of d by the local input dimension $d_{\text{in}(v)}$ highlights the adaptivity of deep networks to sparse compositional structure and clarifies how hierarchical representations alleviate the curse of dimensionality. The compositional level L contributes to the rate through both the depth D and the denominator in the approximation exponent. This reflects the cost of propagating approximation errors through multiple compositions. Nevertheless, the dependence on L is substantially milder than the ambient-dimensional dependence in classical approximation rates. We provide explicit rates in Remark 1 to make this phenomenon concrete. However, the optimality of these rates in the norm-constrained setting is unclear, and establishing matching lower bounds remains an important direction for future work.

Comparison with prior work To the best of our knowledge, this is the first result establishing approximation rates for sparse compositional functions in terms of a norm constraint. In contrast, existing works analyze compositional function classes with guarantees expressed in terms of parameter counts M. In particular, [72, 13, 49] study compositional Hölder spaces using deep neural networks and obtain rates of order $\max_v M^{-\alpha^*_v / d_{\text{in}(v)}}$. The same rate is derived for ridge functions (single-index models) with Lipschitz continuity using deep CNNs [95, 30]. Moreover, Mao et al. [56] investigate compositional polynomial spaces with deep CNNs. While all these works demonstrate that the rate depends exponentially on intermediate latent dimensions rather than the ambient dimension, their results are formulated exclusively in terms of parameter counts.

To clarify how the rate scales with dimension, we instantiate the bounds with representative choices of the level L studied in prior work.

Remark 1 (Explicit approximation rates examples)**.** *We consider the following three examples.*

(Multi-index model) *Consider the multi-index model* $f(x) = g(Ax)$ *with* $A \in \mathbb{R}^{s\times d}$ *and* $s < d$ *[54]. This induces a two-level compositional structure with* $L = 2$, *where* $d_{in(v^{(1)})} = d$ *for all* $v^{(1)} \in \mathcal{V}^{(1)}$ *(linear projection nodes) and* $d_{in(v^{(2)})} = s$ *for the nonlinear node. Although this structure is not*

sparse compositional, it still fits within our framework. The approximation rate becomes

$$\max_{v^{(\ell)}:\pi_{1\to 2}} K^{-\frac{2\alpha^*_{v^{(\ell)}}}{4+(4\lceil\log_2(s+r)\rceil+6)d_{in(v^{(\ell)})}}} \lesssim K^{-\frac{\alpha}{C_1 s}},$$

where $\alpha = r + \beta$ denotes the regularity of g and $C_1 = 2\lceil\log_2(s+r)\rceil + 5$ is independent of the ambient dimension d. Thus, the rate depends solely on the intrinsic dimension s, demonstrating the adaptivity of overparameterized networks.

(Binary tree) *In the binary tree structure [68], $L = \log_2 d$ and $d_{in(v)} = 2$. The approximation bound is*

$$\|f - \phi\|_\infty \le C_{\mathcal{G}}^{\mathrm{app}} \max_{v^{(\ell)}:\pi_{1\to L}} K^{-\alpha^*_{v^{(\ell)}}/(C_2 \log_2 d)},$$

where $C_2 = 2\max_{v\in\mathcal{V}}\lceil\log_2(2+r_v)\rceil + 4$ is independent of d. The exponent depends on d only through $\log_2 d$. For fixed bivariate smoothness bounds, let $\overline{D} = 2\max_v\lceil\log_2(2+r_v)\rceil + 2$ and $\alpha_{\max} = \max_v \alpha_v$. The constant $C_{\mathcal{G}}^{\mathrm{app}}$ can be bounded by

$$C_{\mathcal{G}}^{\mathrm{app}} \le C_0(\log_2 d)(d/2+1)^\rho, \qquad \rho = \frac{\alpha_{\max}(\overline{D}+1)}{2\overline{D}+4},$$

where C_0 depends on the fixed local smoothness and normalization bounds. The remaining dependence is polynomial in d. Overall, the explicit dependence on d is polynomial in the prefactor and logarithmic in the denominator of the rate exponent, with no exponential factor in d.

(General constant level) *Alternatively, one may take $L = c$ as a constant independent of d [72]. Then the rate scales as*

$$\max_{v^{(\ell)}:\pi_{1\to L}} K^{-\frac{2\alpha^*_{v^{(\ell)}}}{2c+\left(2c\max_v\left\lceil\log_2(d_{in(v)}+r_v)\right\rceil+3c\right)d_{in(v^{(\ell)})}}} \lesssim \max_{v^{(\ell)}:\pi_{1\to L}} K^{-\frac{\alpha^*_{v^{(\ell)}}}{C_3 d_{in(v^{(\ell)})}}},$$

where $C_3 = c\max_v\left\lceil\log_2(d_{in(v)}+r_v)\right\rceil + 2.5$ is also independent of d. In this regime, the rate depends only on the intermediate dimensions $d_{in(v)}$, which are typically small, and is independent of the ambient dimension d. In this regime, the displayed exponent depends on the local dimensions, while the prefactor $C_{\mathcal{G}}^{\mathrm{app}}$ still records the size and structure of the DAG.

4 Statistical analysis for sparse compositional functions

The goal of supervised learning with least squares loss is to estimate a minimizer of the expected risk

$$\mathcal{R}(f) := \mathbb{E}_\rho[(f(x)-y)^2] = \int_{X\times\mathbb{R}} (f(x)-y)^2 \mathrm{d}\rho(x,y)$$

over all measurable functions, based on a training sample set $\{(x_i,y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \rho^n$. Here ρ is the unknown joint probability distribution of $X\times\mathbb{R}$. The quality of a learning solution $\hat{f}$ is measured by the excess risk

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f_*),$$

where f_* denotes a minimizer of $\mathcal{R}(f)$. It quantifies the statistical error of $\hat{f}$ relative to the optimal predictor under the population distribution ρ, rather than on the training sample. In this section, we establish excess risk bounds for the empirical risk minimizer $\hat{f}$ defined via empirical risk minimization

$$\hat{f} := \operatorname*{arg\,min}_{f\in\mathcal{NN}_B(W,D,K)} \widehat{\mathcal{R}}(f), \quad \widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^n (f(x_i)-y_i)^2. \tag{5}$$

Here, $\mathcal{NN}_B(W,D,K)$ denotes neural networks with outputs uniformly bounded by $B>0$. Such realizations are obtained by composing the network from Theorem 2 with the clipping operator $\chi_B : \mathbb{R}\to[-B,B]$, given by $\chi_B(x) = \sigma(x) - \sigma(-x) - \sigma(x-B) + \sigma(-x-B)$. This ensures bounded outputs and enables moment control; without loss of generality, we take $B=1$.

Assumption 2. *The input space $\mathcal{X}\subset\mathbb{R}^d$ satisfies $\|x\|\le 1$, and the output satisfies $|y|\le B$ for some $B>0$. Without loss of generality, we may assume $B=1$ by rescaling.*

The input constraint $\|x\| \leq 1$ can be enforced by rescaling for bounded inputs. The boundedness of the outputs is standard and ensures moment control for applying concentration inequalities.

The following theorem establishes the sample complexity of learning sparse compositional Hölder functions with neural networks under Frobenius norm constraints. The proof is provided in Appendix C.

Theorem 3. *Let $f_* : [-1,1]^d \to \mathbb{R}$ be a sparse compositional function defined by* (4)*, associated with a DAG $(\mathcal{V}, \mathcal{E})$ with L levels. Under Assumptions 1 and 2, for any $n \in \mathbb{N}_+$, there exists an estimator $\hat{f} \in \mathcal{NN}_1(W, D, K)$ with depth $D = 2L \max_{\ell, v^{(\ell)}} \lceil \log_2(d_{in(v^{(\ell)})} + r_{v^{(\ell)}}) \rceil + 2L$, such that, for any width $W \geq \max_{\ell, v^{(\ell)}} K^{\frac{2d_{in(v^{(\ell)})} + \alpha_{v^{(\ell)}}}{(D+L)\, d_{in(v^{(\ell)})} + 2L}}$, and norm constraint*

$$K \asymp \max_{v^{(\ell)}: \pi_{1\to L}} (n)^{\frac{1}{2} \frac{2L + (D+L) d_{in(v^{(\ell)})}}{2L + (D+L) d_{in(v^{(\ell)})} + 4\alpha^*_{v^{(\ell)}}}}, \tag{6}$$

the excess risk satisfies, with probability at least $1 - \delta$,

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f_*) \lesssim \sqrt{\log \frac{4}{\delta}} \max_{v^{(\ell)}: \pi_{1\to L}} n^{-\frac{2\alpha^*_{v^{(\ell)}}}{2L + (D+L) d_{in(v^{(\ell)})} + 4\alpha^*_{v^{(\ell)}}}},$$

where $\pi_{1\to L}$ denotes the critical path selected by the error propagation in (17) *from level* 1 *to level L, and $\alpha^*_{v^{(\ell)}} := \alpha_{v^{(\ell)}} \prod_{m=\ell+1}^{L} \min\{\alpha_{v^{(m)}}, 1\}$ is the effective regularity index along this path.*

To the best of our knowledge, this is the first result establishing sample complexity guarantees for estimators generated by neural networks that cover the overparameterized regime for sparse compositional function classes. The depth D scales linearly with the number of compositional levels L, while the width W can be arbitrarily large beyond a minimal threshold, placing the model in the overparameterized regime. The learning rate is governed by the most restrictive node along the critical path and can be further upper-bounded by the worst-case node over all possible paths. The resulting rate depends on local structural complexity rather than the ambient dimension, thereby mitigating the curse of dimensionality. More explicit dimension dependence for representative compositional models is provided in Remark 2. We further discuss this theorem below.

Scaling of the norm constraint K The choice of K in (6) follows from balancing approximation and estimation errors, corresponding to bias induced by hypothesis class restriction and variance arising from finite samples. This trade-off is formalized via a standard error decomposition. Define $\tilde{f} := \arg\min_{f \in \mathcal{NN}_B(W,D,K)} \mathcal{R}(f)$, then the excess risk

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f_*) \leq 2 \sup_{f \in \mathcal{NN}_1(W,D,K)} \left| \mathcal{R}(f) - \widehat{\mathcal{R}}(f) \right| + \mathcal{R}(\tilde{f}) - \mathcal{R}(f_*). \tag{7}$$

Equation (21) provides its proof for completeness. The first term captures the estimation (sample) error, whereas the second term represents the approximation error. Theorem 2 shows that the approximation error decays polynomially as $K^{-\gamma}$, where γ depends on the singularity index α^*_v and the local input dimensions $d_{\mathrm{in}(v)}$. This rate decreases as K increases, since a larger K corresponds to a richer hypothesis class to approximate the target function. The empirical Rademacher complexity $\widehat{\mathfrak{R}}_n(\mathcal{NN}(W, D, K))$ provides a tool for controlling the estimation error. Indeed, Golowich et al. [35] show that $\widehat{\mathfrak{R}}_n(\mathcal{NN}(W, D, K)) \lesssim K/\sqrt{n}$. Related norm-based bounds have also been studied in [63, 11, 34, 67]. Balancing $K^{-\gamma}$ and $K/\sqrt{n}$ yields the optimal scaling $K \asymp n^{1/(2(\gamma+1))}$, leading to the stated excess risk rate. Thus, K acts as a continuous capacity parameter, analogous to model size in classical bias–variance tradeoffs but better suited to the overparameterized regime.

Comparison with parameter-count-based bounds and beyond Classical excess risk analyses based on parameter count M balance approximation and estimation errors of the form $M^{-\tau}$ and M/n, where $\tau > 0$ depends on the singularity index and local input dimensions. Balancing these two terms yields the optimal choice $M \asymp n^{1/(1+\tau)}$, which is smaller than the sample size n, and leads to the rate $n^{-\tau/(1+\tau)}$. Variants of this approximation–estimation tradeoff appear throughout the theory of deep ReLU networks for nonparametric regression and compositional function classes [13, 72, 50, 39], as well as in theoretical analyses of deep convolutional networks based on approximation, consistency,

and non-asymptotic excess-risk bounds [96, 32, 48]. Having compared approximation rates in the previous section, we now examine the estimation error in more detail. A key limitation of parameter-count-based bounds is that when $M > n$, the term M/n no longer vanishes, and thus fails to capture the overparameterized regime [10, 16]. Moreover, achieving fast rates of order n^{-1} typically requires refined techniques such as local Rademacher complexity and variance-dependent bounds [9, 78, 86]. This raises the question of whether similar refined analyses can be developed for norm-constrained neural networks. While such results exist for infinite-width shallow models [89, 76], extending localized fast-rate analyses to deep norm-constrained networks in the overparameterized regime remains largely open. Our rates exhibit the same qualitative dependence on smoothness and intrinsic dimension as classical results, but their optimality in the norm-constrained setting is still unclear. For a direct comparison, the classical minimax squared-risk benchmark for a d-dimensional Hölder ball is $n^{-2\alpha/(2\alpha+d)}$ [38]. Specializing Theorem 3 to one compositional level gives the slower upper bound $n^{-2\alpha/\{2+d(D+1)+4\alpha\}}$. For compositional classes, parameter-count analyses obtain rates governed by $n^{-2\alpha_v^*/(2\alpha_v^*+d_{\mathrm{in}(v)})}$, up to logarithmic factors and under their respective assumptions [72, 49]. Our contribution is the control of approximation by a product norm in a class permitting arbitrarily large width; the estimation argument uses existing Rademacher bounds. Neither part is known to be sharp, and the resulting upper bound is not a minimax characterization. Matching lower bounds and localized complexity analyses remain open.

Remark 2 (Explicit rates in representative settings)**.** *We specialize the excess risk bound to several representative compositional structures considered above with the same constants in Remark 1.*

(Multi-index model) *For the multi-index model $f(x) = g(Ax)$ with intrinsic dimension s ($L = 2$), the rate becomes*

$$n^{-\frac{1}{2}\frac{2\alpha}{C_1 s+2\alpha}},$$

where α is the regularity of g and C_1 is independent of the ambient dimension d.

(Binary tree) *For the binary tree structure with $L = \log_2 d$ and $d_{in(v)} = 2$, the rate is*

$$\max_{v^{(\ell)}:\pi_{1\to L}} n^{-\frac{1}{2}\frac{2\alpha^*_{v^{(\ell)}}}{C_2 \log_2 d+2\alpha^*_{v^{(\ell)}}}},$$

so the dimensional dependence enters only through $\log_2 d$.

(General constant level) *For constant level $L = c$, the rate becomes*

$$\max_{v^{(\ell)}:\pi_{1\to L}} n^{-\frac{1}{2}\frac{2\alpha^*_{v^{(\ell)}}}{C_3 d_{in(v^{(\ell)})}+2\alpha^*_{v^{(\ell)}}}}.$$

In all cases, the rates depend on the local dimension, indicating that hierarchical representations mitigate the curse of dimensionality even in the overparameterized regime.

4.1 Discussion

Compositional and mixed smoothness perspectives Classical approximation and excess risk rates, $M^{-\alpha/d}$ and $n^{-2\alpha/(2\alpha+d)}$, depend on the interplay between smoothness α and dimension d [65, 38], reflecting the blessing of smoothness and the curse of dimensionality [29]. This motivates two approaches to adaptivity in high dimensions: exploiting structure in dimensionality or in regularity. We focus on the former via the compositional principle, showing the rates depend on local dimensionality (see Remarks 1 and 2). A complementary line studies function spaces with mixed derivatives, which limit high-order interactions across variables [60, 79]. Most results in this direction are based on parameter-count analyses and demonstrate mitigation of the curse of dimensionality. It remains unclear whether norm-constrained deep networks can exploit such structure or achieve generalization guarantees.

Explicit and implicit regularization The analysis in this work is based on norm-constrained empirical risk minimization (5), corresponding to an Ivanov regularization scheme. In contrast, modern neural networks are often trained in the overparameterized regime without explicit constraints, achieving interpolation while still generalizing well, a phenomenon known as benign overfitting [12, 15]. This behavior is commonly interpreted through the lens of implicit regularization, which

can be idealized as $\min_\theta \ \kappa(\theta)$ such that $f_\theta(x_i) = y_i$, see, e.g., [62, 77, 37]. A classical alternative is Tikhonov regularization, $\min_{f \in \mathcal{NN}(W,D)} \widehat{\mathcal{R}}(f) + \lambda\kappa(\theta), \lambda > 0$, which introduces a norm penalty in the objective and provides a natural bridge between explicit regularization and implicit bias. Understanding the convergence rates of Tikhonov-regularized estimators may provide a principled route to characterizing minimal-norm interpolation and implicit regularization in deep networks. Establishing such connections in overparameterized settings remains largely open.

5 Conclusion

We provide a norm-based perspective on the compositional principle in deep neural networks, showing that both approximation and generalization can be characterized directly in terms of Frobenius norm constraints, beyond classical parameter-count-based analyses. Our results demonstrate that, for sparse compositional functions represented by DAGs, the learning rates are governed by local smoothness and intrinsic input dimensions along critical paths. In particular, the dependence on the ambient dimension is replaced by logarithmic or intermediate dimensions in representative compositional models, thereby mitigating the CoD. These findings extend to the overparameterized regime, where network width can be arbitrarily large beyond a minimal threshold. Overall, this work highlights that norm-constrained deep networks can exploit compositional structure to achieve statistically efficient learning in high dimensions.

Acknowledgments

L. R. acknowledges the financial support of the European Commission (Horizon Europe grant ELIAS 101120237), and the Ministry of Education, University and Research (FARE grant ML4IP R205T7J2KP). T. P. acknowledges the support from AFOSR project (FA955024-1-0231).

References

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

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Mazen Ali and Anthony Nouy. Approximation theory of tree tensor networks: Tensorized univariate functions. *Constructive Approximation*, 58(2):463–544, 2023.

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

[5] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

[6] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263. PMLR, 2018.

[7] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

[8] Markus Bachmayr, Anthony Nouy, and Reinhold Schneider. Approximation by tree tensor networks in high dimensions: Sobolev and compositional functions. *arXiv preprint arXiv:2112.01474*, 2021.

[9] Peter Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

[10] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[11] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[12] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[13] Benedikt Bauer and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4):2261–2285, 2019.

[14] Eric Baum and David Haussler. What size net gives valid generalization? *Advances in Neural Information Processing Systems*, 1, 1988.

[15] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

[16] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116 (32):15849–15854, 2019.

[17] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[18] Alberto Bietti, Joan Bruna, and Loucas Pillaud-Vivien. On learning gaussian multi-index models with gradient flow. *arXiv preprint arXiv:2310.19793*, 2023.

[19] Francesco Cagnetta, Leonardo Petrini, Umberto M Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14 (3):031001, 2024.

[20] Yunlu Chen, Yang Li, Keli Liu, and Feng Ruan. Kernel learning in ridge regression "automatically" yields exact low rank solution. *arXiv preprint arXiv:2310.11736*, 2023.

[21] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2 (3):113–124, 1956.

[23] Wolfgang Dahmen. Compositional sparsity, approximation classes, and parametric transport equations. *Constructive Approximation*, 61(2):219–283, 2025.

[24] Alex Damian, Loucas Pillaud-Vivien, Jason Lee, and Joan Bruna. Computational-statistical gaps in gaussian single-index models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1262–1262. PMLR, 2024.

[25] Yatin Dandi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The computational advantage of depth: Learning high-dimensional hierarchical functions with gradient descent. *arXiv preprint arXiv:2502.13961*, 2025.

[26] David A Danhofer, Davide D'Ascenzo, Rafael Dubach, and Tomaso Poggio. Position: A theory of deep learning must include compositional sparsity. *arXiv preprint arXiv:2507.02550*, 2025.

[27] Leonardo Defilippis, Florent Krzakala, Bruno Loureiro, and Antoine Maillard. Optimal scaling laws in learning hierarchical multi-index models. *arXiv preprint arXiv:2602.05846*, 2026.

[28] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

[29] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[30] Zhiying Fang, Han Feng, Shuo Huang, and Ding-Xuan Zhou. Theory of deep convolutional neural networks ii: Spherical analysis. *Neural Networks*, 131:154–162, 2020.

[31] Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[32] Han Feng, Shuo Huang, and Ding-Xuan Zhou. Generalization analysis of cnns for classification on spheres. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6200–6213, 2023.

[33] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, pages 1871–1905, 2009.

[34] Tomer Galanti, Mengjia Xu, Liane Galanti, and Tomaso Poggio. Norm-based generalization bounds for compositionally sparse neural networks. *arXiv preprint arXiv:2301.12033*, 2023.

[35] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *Information and Inference: A Journal of the IMA*, 9(2):473–504, 2020.

[36] Kalanit Grill-Spector and Rafael Malach. The human visual cortex. *Annu. Rev. Neurosci.*, 27(1):649–677, 2004.

[37] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[38] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[39] Zhi Han, Siquan Yu, Shao-Bo Lin, and Ding-Xuan Zhou. Depth selection for deep relu nets in feature extraction and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 1853–1868, 2020.

[40] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

[41] Shuo Huang, Hippolyte Labarrière, Ernesto De Vito, Tomaso Poggio, and Lorenzo Rosasco. Learning multi-index models with hyper-kernel ridge regression. *arXiv preprint arXiv:2510.02532*, 2025.

[42] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[43] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

[44] Yuling Jiao, Yang Wang, and Yunfei Yang. Approximation bounds for norm constrained neural networks with applications to regression and gans. *Applied and Computational Harmonic Analysis*, 65:249–278, 2023.

[45] Anatoli B Juditsky, Oleg Lepski, and Alexandre B Tsybakov. Nonparametric estimation of composite functions. *Annals of Statistics*, 37(3):1360–1404, 2009.

[46] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[47] T Klock, A Lanteri, and S Vigogna. Estimating multi-index models with response-conditional least squares. *Electronic Journal of Statistics*, 15(1):589–629, 2021.

[48] Michael Kohler and Benjamin Kohler. Analysis of convolutional neural network image classifiers in a rotationally symmetric model. *IEEE Transactions on Information Theory*, 69(8):5203–5218, 2023.

[49] Michael Kohler and Sophie Langer. On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics*, 49(4):2231–2249, 2021.

[50] Michael Kohler, Adam Krzyżak, and Sophie Langer. Estimation of a function of low local dimensionality by deep neural networks. *IEEE Transactions on Information Theory*, 68(6):4032–4042, 2022.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[52] Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, volume 4, pages 950–957. Morgan Kaufmann, 1991.

[53] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*, volume 23. Springer Science & Business Media, 1991.

[54] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[55] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

[56] Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Approximating functions with multi-features by deep convolutional neural networks. *Analysis and Applications*, 21(01):93–125, 2023.

[57] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on learning theory*, pages 2388–2464. PMLR, 2019.

[58] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[59] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[60] Hadrien Montanelli and Qiang Du. New error bounds for deep relu networks using sparse grids. *SIAM Journal on Mathematics of Data Science*, 1(1):78–92, 2019.

[61] Thomas Nagler and Sophie Langer. Optimal neural network approximation of smooth compositional functions on sets with low intrinsic dimension. *arXiv preprint arXiv:2602.03539*, 2026.

[62] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[63] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015.

[64] Rahul Parhi and Robert D Nowak. Near-minimax optimal estimation with shallow relu neural networks. *IEEE Transactions on Information Theory*, 69(2):1125–1140, 2022.

[65] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

[66] Tomaso Poggio. On efficiently computable functions, deep networks and sparse compositionality. *arXiv preprint arXiv:2510.11942*, 2025.

[67] Tomaso Poggio and Maia Fraser. Compositional sparsity of learnable functions. *Bulletin of the American Mathematical Society*, 61(3):438–456, 2024.

[68] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

[69] Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383 (6690):1461–1467, 2024.

[70] Levi Rauchwerger, Stefanie Jegelka, and Ron Levie. Dense neural networks are not universal approximators. *arXiv preprint arXiv:2602.07618*, 2026.

[71] Yunwei Ren, Yatin Dandi, Florent Krzakala, and Jason D Lee. Provable learning of random hierarchy models and hierarchical shallow-to-deep chaining. *arXiv preprint arXiv:2601.19756*, 2026.

[72] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

[73] Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1):e2408799121, 2025.

[74] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *arXiv preprint arXiv:1906.05497*, 2019.

[75] Zhongjie Shi, Zhiying Fang, and Yuan Cao. Approximation and estimation capability of vision transformers for hierarchical compositional models. *Applied and Computational Harmonic Analysis*, page 101849, 2025.

[76] Jonathan W Siegel and Jinchao Xu. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks. *Foundations of Computational Mathematics*, 24(2):481–537, 2024.

[77] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

[78] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[79] Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

[80] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[81] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. Ieee, 2015.

[82] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[83] David C Van Essen and John HR Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in neurosciences*, 6:370–375, 1983.

[84] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[86] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[87] Yahong Yang and Juncai He. Deeper or wider: A perspective from optimal generalization error with sobolev loss. In *International Conference on Machine Learning*, pages 56109–56138. PMLR, 2024.

[88] Yahong Yang, Haizhao Yang, and Yang Xiang. Nearly optimal vc-dimension and pseudo-dimension bounds for deep neural network derivatives. *Advances in Neural Information Processing Systems*, 36: 21721–21756, 2023.

[89] Yunfei Yang and Ding-Xuan Zhou. Nonparametric regression using over-parameterized shallow relu neural networks. *Journal of Machine Learning Research*, 25(165):1–35, 2024.

[90] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[91] Bohan Zhang, Zihao Wang, Hengyu Fu, and Jason D Lee. Neural networks learn generic multi-index models near information-theoretic limit. *arXiv preprint arXiv:2511.15120*, 2025.

[92] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[93] Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

[94] Yuanhe Zhang, Ilja Kuzborskij, Jason D Lee, Chenlei Leng, and Fanghui Liu. Dag-math: Graph-guided mathematical reasoning in llms. *arXiv preprint arXiv:2510.19842*, 2025.

[95] Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124: 319–327, 2020.

[96] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.

A Basic properties of Frobenius norm-constrained neural networks

For any function $f \in \mathcal{NN}(W, D, K)$, the following proposition shows that all intermediate layers can be rescaled with a unit Frobenius norm, while the Frobenius norm of the final layer remains bounded by K, without changing the realized function. This property follows from the positive homogeneity of the ReLU activation and simplifies the subsequent proofs.

Proposition 1 (Rescaling with Frobenius norm)**.** *Every $\phi \in \mathcal{NN}(W, D, K)$ can be written in the form* (1) *such that*

$$\|A_D\|_F \le K, \quad \sqrt{\|A_\ell\|_F^2 + \|b_\ell\|_2^2} \le 1, \quad \textit{for all } \ell = 0, \dots, D-1.$$

Proof of Proposition 1. Let $\phi \in \mathcal{NN}(W, D, K)$ have parameters $\theta = ((A_\ell, b_\ell)_{\ell=0}^{D-1}, A_D)$ satisfying

$$\kappa(\theta) = \|A_D\|_F \prod_{\ell=0}^{D-1} \sqrt{\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1} \le K.$$

Set $s_{-1} := 1$ and recursively define

$$s_\ell := \sqrt{s_{\ell-1}^2 \|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1}, \qquad \ell = 0, \dots, D-1.$$

In particular, $s_\ell \ge 1$. Define the rescaled parameters by

$$\widehat{A}_\ell := \frac{s_{\ell-1}}{s_\ell} A_\ell, \qquad \widehat{b}_\ell := \frac{b_\ell}{s_\ell}, \qquad \ell = 0, \dots, D-1,$$

and set $\widehat{A}_D := s_{D-1} A_D$. These rescalings preserve the width and depth of the network.

To verify that the realized function is unchanged, observe that positive homogeneity of ReLU gives, for every vector z of the appropriate dimension,

$$\sigma\left(\widehat{A}_\ell \frac{z}{s_{\ell-1}} + \widehat{b}_\ell\right) = \sigma\left(\frac{A_\ell z + b_\ell}{s_\ell}\right) = \frac{1}{s_\ell} \sigma(A_\ell z + b_\ell).$$

Since $s_{-1} = 1$, applying this identity successively through the hidden layers shows that the output of the last hidden layer is divided by s_{D-1}. This factor is canceled by $\widehat{A}_D = s_{D-1} A_D$, so the rescaled network realizes the same function ϕ.

For every $\ell = 0, \dots, D-1$, the definition of s_ℓ yields

$$\|\widehat{A}_\ell\|_F^2 + \|\widehat{b}_\ell\|_2^2 = \frac{s_{\ell-1}^2 \|A_\ell\|_F^2 + \|b_\ell\|_2^2}{s_\ell^2} = 1 - \frac{1}{s_\ell^2} \le 1.$$

Moreover, because $s_{\ell-1} \ge 1$,

$$s_\ell^2 = s_{\ell-1}^2 \|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1 \le s_{\ell-1}^2 (\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1).$$

Taking square roots and iterating from $s_{-1} = 1$, we obtain

$$s_{D-1} \le \prod_{\ell=0}^{D-1} \sqrt{\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1}.$$

Consequently,

$$\|\widehat{A}_D\|_F = s_{D-1} \|A_D\|_F \le \kappa(\theta) \le K.$$

Thus, the rescaled parameters satisfy all the asserted norm bounds. □

The following proposition collects several elementary properties of neural networks under Frobenius norm constraints that will be used repeatedly throughout the paper. Analogous properties for alternative choices of $\kappa(\theta)$ have been established in prior work, such as [72, 44].

Proposition 2 (Elementary properties of $\mathcal{NN}(W, D, K)$)**.** *Let $\phi_1 \in \mathcal{NN}(W_1, D_1, K_1)$ and $\phi_2 \in \mathcal{NN}(W_2, D_2, K_2)$.*

(2.1) (Linear combination) Suppose $D_1 = D_2 = D$, for any constants $c_1, c_2 \in \mathbb{R}$, the function $\phi = c_1\phi_1 + c_2\phi_2$ satisfies

$$\phi \in \mathcal{NN}\left(W_1 + W_2, D, (\sqrt{3})^D \sqrt{(c_1K_1)^2 + (c_2K_2)^2}\right).$$

More generally, the linear combination of N neural networks with same depth satisfies $\phi \in \mathcal{NN}\left(\sum_{i=1}^N W_i, D, \sqrt{N+1}^D \sqrt{\sum_{i=1}^N (c_iK_i)^2}\right)$.

(2.2) (Concatenation) If $D_1 = D_2 = D$, define the concatenation $\phi(x) := (\phi_1(x), \phi_2(x))$, then

$$\phi \in NN\big(W_1 + W_2, D, (\sqrt{3})^D \sqrt{K_1^2 + K_2^2}\big).$$

More generally, the concatenation for N neural networks with same depth satisfies $\phi \in \mathcal{NN}\left(\sum_{i=1}^N W_i, D, \sqrt{N+1}^D \sqrt{\sum_{i=1}^N K_i^2}\right)$.

(2.3) (Composition) If $d_2 = k_1$, then the composition

$$\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}\left(\max\{W_1, W_2\}, D_1 + D_2, \sqrt{2}^{D_1} K_2 \sqrt{K_1^2 + 2}\right).$$

More generally, the composition of N neural networks with aligned input and output dimensions satisfies $\phi \in \mathcal{NN}\left(\max_i\{W_i\}, \sum_{i=1}^N D_i, \sqrt{2}^{\sum_{i=1}^{N-1} D_i} K_N \prod_{i=1}^{N-1} \sqrt{K_i^2 + 2}\right)$.

Since the Frobenius norm constraint is defined through the product over D layers, the factors involving exponents of D arise naturally. This also explains why our constructions favor shallower but wider networks.

Proof of Proposition 2. (Proof of Proposition 2.1) Let $\phi_1 \in \mathcal{NN}(W_1, D, K_1)$ and $\phi_2 \in \mathcal{NN}(W_2, D, K_2)$ be two ReLU networks of the same depth D. Write their realizations as

$$\phi_i(x) = A_D^{(i)}\sigma\big(A_{D-1}^{(i)}\sigma(\cdots\sigma(A_0^{(i)}x + b_0^{(i)})\cdots) + b_{D-1}^{(i)}\big), \quad i = 1, 2.$$

By Proposition 1, we have

$$\|A_\ell^{(i)}\|_F^2 + \|b_\ell^{(i)}\|_2^2 \le 1, \quad \ell = 0, \ldots, D-1.$$

We construct a network that computes ϕ_1 and ϕ_2 in parallel. For each hidden layer $\ell = 0, \ldots, D-1$, define the block-diagonal augmented matrix

$$A_\ell = \begin{pmatrix} A_\ell^{(1)} & 0 \\ 0 & A_\ell^{(2)} \end{pmatrix}, \quad b_\ell = \begin{pmatrix} b_\ell^{(1)} \\ b_\ell^{(2)} \end{pmatrix}. \tag{8}$$

This network has width $W_1 + W_2$ and depth D, and satisfies

$$\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1 = \|A_\ell^{(1)}\|_F^2 + \|b_\ell^{(1)}\|_2^2 + \|A_\ell^{(2)}\|_F^2 + \|b_\ell^{(1)}\|_2^2 + 1 \le 3.$$

At the output layer, define

$$A_D := \big(c_1A_D^{(1)},\ c_2A_D^{(2)}\big),$$

so that the resulting network computes $\phi(x) = c_1\phi_1(x) + c_2\phi_2(x)$. Moreover,

$$\|A_D\|_F^2 \le (c_1K_1)^2 + (c_2K_2)^2.$$

Combining the above bounds over all layers yields

$$\phi \in NN\Big(W_1 + W_2, D, (\sqrt{3})^D \sqrt{(c_1K_1)^2 + (c_2K_2)^2}\Big).$$

Moreover, the linear combination of N neural networks yields hidden-layer matrices A_ℓ and bias vectors b_ℓ with N blocks as (8). Combining this block construction with the rescaling proposition,

we have $\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1 \le N + 1$. The output matrix is given by $A_D := \big(c_1 A_D^{(1)}, \dots, c_N A_D^{(N)}\big)$. Combining these estimates with the definition of $\kappa(\theta)$ completes the proof.

(Proof of Proposition 2.2) We present the proof for the concatenation of two neural networks; the general case of N networks follows by the same argument. construct a network $\phi = (\phi_1, \phi_2)$ by running the two networks in parallel. For each hidden layer $\ell = 0, \dots, D-1$, the augmented matrix is block-diagonal and given by

$$A_\ell := \begin{pmatrix} A_\ell^{(1)} & 0 \\ 0 & A_\ell^{(2)} \end{pmatrix}, \quad b_\ell = \begin{pmatrix} b_\ell^{(1)} \\ b_\ell^{(2)} \end{pmatrix}$$

and the final layer is

$$A_D := \begin{pmatrix} A_D^{(1)} & 0 \\ 0 & A_D^{(2)} \end{pmatrix}.$$

The resulting network has depth D and width $W_1 + W_2$, and clearly realizes $\phi(x) = (\phi_1(x), \phi_2(x))$.

Moreover, for $\ell = 0, \dots, D-1$, the rescaling proposition gives

$$\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1 = \|A_\ell^{(1)}\|_F^2 + \|b_\ell^{(1)}\|_2^2 + \|A_\ell^{(2)}\|_F^2 + \|b_\ell^{(2)}\|_2^2 + 1 \le 3.$$

For the last layer,

$$\|A_D\|_F^2 = \|A_D^{(1)}\|_F^2 + \|A_D^{(2)}\|_F^2 \le K_1^2 + K_2^2.$$

Therefore,

$$\|A_D\|_F \prod_{\ell=0}^{D-1} \sqrt{\|A_\ell\|_F^2 + \|b_\ell\|_2^2} \le (\sqrt{3})^D \sqrt{K_1^2 + K_2^2}.$$

The extension to N networks follows by the same parallel construction with an N-block diagonal matrix at each layer, giving the factor $(\sqrt{N+1})^D$ and the bound $K = (\sqrt{N+1})^D \sqrt{\sum_{i=1}^N K_i^2}$.

(Proof of Proposition 2.3) Assume $d_2 = k_1$ so that the composition $\phi = \phi_2 \circ \phi_1$ is well-defined. We construct the composed network by stacking the layers of ϕ_2 after those of ϕ_1. The resulting network has depth $D_1 + D_2$ and width at most $\max\{W_1, W_2\}$. For $\ell = 0, \dots, D_1 - 1$, we keep the layers of ϕ_1, and append the layers of ϕ_2 afterward. The key step is the interface layer. At the transition, the mapping becomes

$$x \mapsto \sigma\big(A_0^{(2)} \phi_1(x) + b_0^{(2)}\big).$$

At layer D_1, the augmented matrix will be

$$\widetilde{A}_{D_1} = A_0^{(2)} A_{D_1}^{(1)}, \quad \tilde{b}_{D_1} = b_0^{(2)}.$$

Note that $\|\widetilde{A}_{D_1}\|_F^2 + \|\widetilde{b}_{D_1}\|_2^2 + 1 = \left\| \left(A_0^{(2)} A_{D_1}^{(1)},\, b_0^{(2)}, 1 \right) \right\|_F^2$ can be estimated by

$$\left\| \left(A_0^{(2)},\, b_0^{(2)}, 1 \right) \begin{pmatrix} A_{D_1}^{(1)} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\|_F^2 \le \left\| \left(A_0^{(2)},\, b_0^{(2)}, 1 \right) \right\|_F^2 \left\| \begin{pmatrix} A_{D_1}^{(1)} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\|_F^2.$$

By Proposition 1, the Frobenius norm of the composed network satisfies

$$\begin{aligned} \kappa(\theta) &\le \left(\prod_{\ell=0}^{D_1-1} \sqrt{\|A_\ell^{(1)}\|_F^2 + \|b_\ell^{(1)}\|_2^2 + 1} \right) \cdot \sqrt{\|A_{D_1}^{(1)}\|_F^2 + 2} \cdot K_2 \\ &\le 2^{D_1/2} K_2 \sqrt{K_1^2 + 2}. \end{aligned}$$

The general case of N compositions follows by induction and it completes the proof.

□

B Approximation ability of Frobenius norm-constrained neural networks

B.1 Proof of Subsection 3.2

Based on the elementary properties of Frobenius norm-constrained neural networks, we are now ready to establish the approximation rate for Hölder spaces, starting with the approximation of the square function.

Lemma 1 (Approximation of x^2)**.** *For any $k \in \mathbb{N}$, there exists $\phi_k \in \mathcal{NN}(k, 1, 3)$, such that $\phi_k(x) = 0$ for $x \le 0$; $\phi_k(x) \in [0, 1]$ for $x \in [0, 1]$; and $|x^2 - \phi_k(x)| \le \frac{1}{2k^2}$ for $x \in [0, 1]$.*

Our proof follows a similar strategy to Jiao et al. [44], who study approximation rates under the operator norm $\|A\| := \sup_{\|x\|_\infty \le 1} \|Ax\|_\infty$. Although both approaches construct shallow, wide neural networks based on Riemann sum approximations and achieve the same norm bound 3, our construction relies on the Frobenius norm, which is more aligned with practical implementations. Compared with the approximation results in [90], the key differences of our result are as follows. First, Yarotsky constructs an approximation to x^2 using a deep ReLU network via composition of sawtooth-type functions, whereas our Lemma 1 uses a shallow but wide construction. Second, Lemma 1 achieves the approximation rate $\varepsilon \asymp k^{-2}$ with a uniform Frobenius norm bound $K \le 3$ independent of the width k. In contrast, in Yarotsky's deep compositional construction, the accuracy satisfies $\varepsilon \asymp 2^{-2L}$ for depth L, and the resulting multiplicative Frobenius constraint satisfies $K = \prod_{\ell=1}^{L} (2^{-2\ell} \sqrt{2^2 + (-4)^2 + 2^2} \sqrt{3 + \frac{5}{4} + 1}) \lesssim 2^{-L(L+1)/2} 12^L$. Equivalently, $K \asymp \exp\big(-c(\log(1/\varepsilon))^2\big)$ for some constant $c > 0$. Thus, the approximation accuracy and the norm budget are intrinsically coupled in a sub-exponential manner.

Proof of Lemma 1. For $x \in [0, 1]$, note that

$$x^2 = \int_0^x 2(x - b)db = \int_0^1 2\sigma(x - b)db,$$

where $\sigma(t) = \max\{0, t\}$ denotes the ReLU activation. Fix $k \in \mathbb{N}$ and define

$$\phi_k(x) := \frac{1}{k} \sum_{i=1}^{k} 2\sigma\left(x - \frac{2i - 1}{2k}\right), \quad x \in \mathbb{R}. \tag{9}$$

The function ϕ_k is the midpoint Riemann sum approximation of the above integral using k equally spaced points in $[0, 1]$. Clearly, $\phi_k(x) = 0$ for $x \le 0$, and $\phi_k(x) \in [0, 1]$ for all $x \in [0, 1]$. Following the proof of [44, Lemma 3.3], the approximation error can be bounded by

$$\sup_{x \in [0,1]} |x^2 - \phi_k(x)| \le \frac{1}{2k^2}.$$

The function ϕ_k can be realized by a ReLU network with one hidden layer and width k. The corresponding parameters are

$$A_0 = (1, \dots, 1)^\top \in \mathbb{R}^{k \times 1}, \quad b_0 = \left(-\frac{2i - 1}{2k}\right)_{i=1}^{k} \in \mathbb{R}^k, \quad A_1 = \left(\frac{2}{k}, \dots, \frac{2}{k}\right) \in \mathbb{R}^{1 \times k},$$

The above construction satisfies

$$\|A_1\|_F = \frac{2}{\sqrt{k}},$$

and since $\sum_{i=1}^{k} (2i - 1)^2 = \sum_{i=1}^{k} 4i^2 + \sum_{i=1}^{k} 1 - \sum_{i=1}^{k} 4i = 4 \cdot \frac{k(k+1)(2k+1)}{6} - 4 \cdot \frac{k(k+1)}{2} + k = \frac{4k^3 - k}{3}$, there holds

$$\|A_0\|_F^2 + \|b_0\|_2^2 + 1 = k + \frac{1}{4k^2} \sum_{i=1}^{k} (2i - 1)^2 + 1 = \frac{4k}{3} - \frac{1}{12k} + 1,$$

Therefore,

$$\kappa(\theta) = \frac{2}{\sqrt{k}} \sqrt{\frac{16k^2 + 12k - 1}{12k}}.$$

Since this quantity is strictly decreasing for all $k \ge 1$, it follows that $\kappa(\theta) \le 3$. Hence, one may choose $K = 3$ independently of the width k, which completes the proof. □

Note that multiplication can be expressed as a linear combination of square functions, namely $xy = 2\big(((x+y)/2)^2 - (x/2)^2 - (y/2)^2\big)$. The scaling of $x+y$ ensures that each argument lies in the interval $[-1,1]$. We then approximate xy by applying the linear combination property (Proposition 2.1) together with Lemma 1.

Lemma 2 (Approximation for xy)**.** *For any $k \in \mathbb{N}$, there exists a network $\psi_k \in \mathcal{NN}(6k, 2, 360)$ with $\psi_k : [-1,1]^2 \to [-1,1]$ such that*

$$|xy - \psi_k(x,y)| \le \frac{3}{k^2}, \quad \forall (x,y) \in [-1,1]^2,$$

and $\psi_k(x,y) = 0$ whenever $xy = 0$.

Proof of Lemma 2. Let $\phi_k \in \mathcal{NN}(k,1,3)$ be the one-hidden-layer network from Lemma 1, which satisfies $\phi_k(t) \in [0,1]$ for $t \in [0,1]$, $\phi_k(t) = 0$ for $t \le 0$ and

$$|t^2 - \phi_k(t)| \le \frac{1}{2k^2}, \quad t \in [0,1].$$

Define the even extension $\widetilde{\phi}_k(t) := \phi_k(t) + \phi_k(-t)$ for $t \in [-1,1]$. Then $\widetilde{\phi}_k \in \mathcal{NN}(2k,1,6)$, and for all $t \in [-1,1]$ we have

$$|t^2 - \widetilde{\phi}_k(t)| = \big|t^2 - \phi_k(|t|)\big| \le \frac{1}{2k^2}, \quad \widetilde{\phi}_k(t) \in [0,1].$$

(Network structure). Based on the observation $xy = 2\big(((x+y)/2)^2 - (x/2)^2 - (y/2)^2\big)$, we construct three networks on $[-1,1]^2$,

$$\Psi_1(x,y) := \widetilde{\phi}_k\left(\frac{x+y}{2}\right), \quad \Psi_2(x,y) := \widetilde{\phi}_k\left(\frac{x}{2}\right), \quad \Psi_3(x,y) := \widetilde{\phi}_k\left(\frac{y}{2}\right).$$

Take $\Psi_1(x,y)$ as an example, if $\tilde{\phi}_k(x) = A_1\sigma(A_0x + b_0)$, $A_0 \in \mathbb{R}^{2k\times 1}$, then

$$\Psi_1(x,y) = A_1\sigma\left(A_0\begin{pmatrix}\frac{1}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + b_0\right),$$

then the new $A_0' := A_0\,(1/2\ 1/2) \in \mathbb{R}^{2k\times 2}$ and $\Psi_1 \in \mathcal{NN}(2k,1,K_{\Psi_1})$, where the norm is bounded by

$$K_{\Psi_1} \le \|A_1\|_F\sqrt{\frac{1}{2}\|A_0\|_F^2 + \|b_0\|_F^2 + 1} \le K_{\tilde{\phi}_k} \le 6.$$

Similarly, we have $\Psi_2 \in \mathcal{NN}(2k,1,6)$ and $\Psi_3 \in \mathcal{NN}(2k,1,6)$. Then, set

$$\Psi(x,y) := 2\Big(\Psi_1(x,y) - \Psi_2(x,y) - \Psi_3(x,y)\Big). \tag{10}$$

By Proposition 2.1 (linear combination), its norm constraint is bounded by

$$K_\Psi \le \sqrt{4}\sqrt{3\times(2\times 6)^2} \le 48$$

and $\Psi \in \mathcal{NN}(6k,1,36)$. To make the output in the range of $[-1,1]$, we need an additional layer $\chi(x)$ given by

$$\chi(x) = \sigma(x) - \sigma(-x) - \sigma\left(x-1\right) + \sigma\left(-x-1\right) = (x \vee (-1)) \wedge 1.$$

Here, $a \vee b := \max\{a,b\}$ and $a \wedge b := \min\{a,b\}$. Its norm is bounded by

$$K_\chi = \sqrt{1+1+1+1}\sqrt{1+1+1+1+1+1+1+1} = 2\sqrt{7}.$$

That is, $\chi \in \mathcal{NN}(4,1,2\sqrt{7})$. Finally, define $\psi_k(x,y) = \chi \circ \Psi(x,y)$. By the compositional property (Proposition 2.3), we have

$$K_{\psi_k} \le \sqrt{2}\cdot 2\sqrt{7}\cdot\sqrt{48^2+2} \approx 359.4 \le 360$$

(Approximation error). For $(x,y) \in [-1,1]^2$, letting $a = \frac{x+y}{2}$, $b = \frac{x}{2}$, $c = \frac{y}{2}$, we have $a,b,c \in [-1,1]$ and

$$|xy - \psi_k(x,y)| \le 2\Big(|a^2 - \widetilde{\phi}_k(a)| + |b^2 - \widetilde{\phi}_k(b)| + |c^2 - \widetilde{\phi}_k(c)|\Big) \le 2\cdot 3\cdot\frac{1}{2k^2} = \frac{3}{k^2}.$$

(Vanishing on the coordinate axes). If $xy = 0$, then either $x = 0$ or $y = 0$, and hence $a = \pm c$ or $a = \pm b$. Since $\widetilde{\phi}_k$ is even, it follows from (10) that $\psi_k(x,y) = 0$ whenever $xy = 0$. This completes the proof. □

After establishing the multiplication operator, the next key step is to extend it to d-degree monomials.

Lemma 3 (Approximation of d-degree monomials)**.** *For any $d \geq 2$ and $k \in \mathbb{N}$, there exists $\phi \in \mathcal{NN}(6dk, 2\lceil \log_2 d \rceil, K)$ with $K \leq 722^{\lceil \log_2 d \rceil} \cdot 2^{\frac{7\lceil \log_2 d \rceil^2 - 7\lceil \log_2 d \rceil}{4}}$ such that $\phi : [-1,1]^d \to [-1,1]$, $\phi(x) = 0$ if any $x_i = 0$ and*

$$|x_1 \cdots x_d - \phi(x)| \leq \frac{6d}{k^2}, \quad x = (x_1, \ldots, x_d)^\top \in [-1,1]^d.$$

Proof of Lemma 3. The proof of this lemma relies on the multiplication operator constructed inductively, as well as on the concatenation and combination properties of neural networks. Let $s := \lceil \log_2 d \rceil$ and set $m := 2^s$, so that $d \leq m < 2d$. Extend $x = (x_1, \ldots, x_d) \in [-1,1]^d$ to $\bar{x} \in [-1,1]^m$ by padding with ones:

$$\bar{x}_i = x_i, \quad i = 1, \ldots, d, \quad \bar{x}_i = 1, \quad i = d+1, \ldots, m.$$

Then $\prod_{i=1}^m \bar{x}_i = \prod_{i=1}^d x_i$.

(Binary-tree network construction). Let $\psi_k \in \mathcal{NN}(6k, 2, 360)$ be the bivariate multiplier from Lemma 2, so that $\psi_k : [-1,1]^2 \to [-1,1]$,

$$|uv - \psi_k(u,v)| \leq \frac{3}{k^2}, \quad \forall (u,v) \in [-1,1]^2, \quad \psi_k(u,v) = 0 \text{ whenever } uv = 0.$$

We construct the network as a binary tree employing ψ_k inductively. For each tree level, define vectors $z^{(\ell)} \in [-1,1]^{m/2^\ell}$ by

$$z^{(0)} := \bar{x}, \quad z_j^{(\ell)} := \psi_k\left(z_{2j-1}^{(\ell-1)}, z_{2j}^{(\ell-1)}\right), \quad \ell = 1, \ldots, s, \quad j = 1, \ldots, \frac{m}{2^\ell}.$$

And denote the composition of all the outputs of the ℓ-th level as

$$\Phi_\ell = \left(z_1^{(\ell)}, \ldots, z_{m/2^\ell}^{(\ell)}\right)$$

Set the final output as $\phi(x) := z_1^{(s)}$. By construction, $\phi(x) = 0$ whenever $x_i = 0$ for some i, since at the first level the corresponding pair-product is zero, and this property propagates through all subsequent applications of ψ_k.

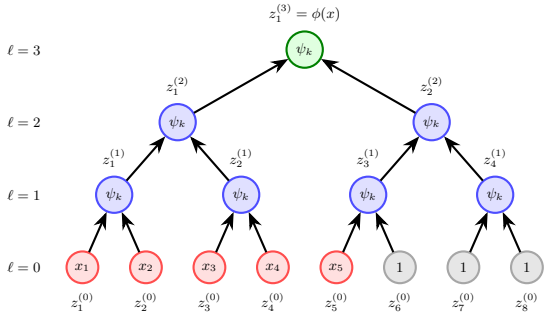

Figure 2: Binary-tree construction of the monomial approximator, illustrated for the case $m = 8$ and $s = 3$. The input is padded with ones so that the number of leaves is $m = 2^s$. Each internal node applies the bivariate multiplier ψ_k to a pair of children, producing the next-level variables $z_j^{(\ell)}$. After s levels, the root outputs $\phi(x) = z_1^{(s)}$, which approximates $\prod_{i=1}^d x_i$.

Observe that the width doubles when two copies of ψ_k are evaluated in parallel; hence the widest width happens when $\ell = 1$, that is, $W = 6k \cdot \frac{m}{2} \leq 6kd$. Moreover, the depth increases by 2 for each level, so that the depth will be $2s = 2\lceil \log_2 d \rceil$. By the concatenation Proposition 2.2, these $m/2^\ell$ parallelization yield a depth-2 network $\Phi_\ell = (\psi_k, \ldots, \psi_k)$ whose norm satisfies

$$K(\Phi_\ell) \leq \left(\sqrt{\frac{m}{2^\ell} + 1}\right)^2 \sqrt{\frac{m}{2^\ell} \cdot 360^2} \leq 720 \left(\frac{m}{2^\ell}\right)^{3/2}.$$

The full network $\phi = \Phi_s \circ \cdots \circ \Phi_1$ is obtained by composing these s levels and each $\Phi_i \in \mathcal{NN}(6kd, 2, 720\left(m/2^i\right)^{3/2})$. Using the composition proposition (2.3), we obtain

$$\begin{aligned} K(\phi) &\leq (\sqrt{2})^{(s-1)2s} \prod_{\ell=1}^{s} \left(K(\Phi_\ell) + 2\right) \\ &\leq 2^{s(s-1)} \prod_{\ell=1}^{s} 722\left(2^{s-\ell}\right)^{3/2} = 722^s 2^{\frac{7}{4}s(s-1)}. \end{aligned}$$

Therefore, the generated neural network $\phi \in \mathcal{NN}(6kd, 2s, 722^s 2^{\frac{7}{4}s(s-1)})$.

(Error bound). Let $P_j^{(\ell)} := \prod_{i=1}^{2^\ell} \bar{x}_{(j-1)2^\ell+i}$ denote the exact block products, and define the uniform error at level ℓ by

$$e_\ell := \max_{1 \leq j \leq m/2^\ell} \left| z_j^{(\ell)} - P_j^{(\ell)} \right|.$$

Clearly $e_0 = 0$. Since all quantities lie in $[-1, 1]$, for each step we have

$$\begin{aligned} &\left| z_j^{(\ell+1)} - P_j^{(\ell+1)} \right| = \left| \psi_k\big(z_{2j-1}^{(\ell)}, z_{2j}^{(\ell)}\big) - P_{2j-1}^{(\ell)} P_{2j}^{(\ell)} \right| \\ &\leq \left| \psi_k\big(z_{2j-1}^{(\ell)}, z_{2j}^{(\ell)}\big) - z_{2j-1}^{(\ell)} z_{2j}^{(\ell)} \right| + \left| z_{2j-1}^{(\ell)} z_{2j}^{(\ell)} - P_{2j-1}^{(\ell)} P_{2j}^{(\ell)} \right| \\ &\leq \frac{3}{k^2} + \left| z_{2j-1}^{(\ell)} - P_{2j-1}^{(\ell)} \right| + \left| z_{2j}^{(\ell)} - P_{2j}^{(\ell)} \right| \leq \frac{3}{k^2} + 2e_\ell, \end{aligned}$$

where we used $|ab - a'b'| \leq |a - a'| + |b - b'|$ for $a, b, a', b' \in [-1, 1]$. Taking the maximum over j yields

$$e_{\ell+1} \leq 2e_\ell + \frac{3}{k^2}.$$

Solving it with $e_0 = 0$ gives

$$e_s \leq \frac{3}{k^2} \sum_{t=0}^{s-1} 2^t = \frac{3(2^s - 1)}{k^2} = \frac{3(m-1)}{k^2} \leq \frac{3m}{k^2} \leq \frac{6d}{k^2},$$

since $m < 2d$. Noting that $P_1^{(s)} = \prod_{i=1}^{m} \bar{x}_i = \prod_{i=1}^{d} x_i$, we obtain

$$|x_1 \cdots x_d - \phi(x)| = |P_1^{(s)} - z_1^{(s)}| \leq e_s \leq \frac{6d}{k^2}, \quad x \in [-1, 1]^d.$$

This completes the proof. □

We are now ready to prove the approximation result for Hölder spaces.

Proof of Theorem 1. The proof is divided into three steps.

Step 1: Approximation based on local Taylor expansion $p(\mathbf{x})$. First, define

$$\psi(t) = \sigma(1 - |t|) = \sigma\big(1 - \sigma(t) - \sigma(-t)\big) \in [0, 1].$$

This function is a "hat function" supported on $[-1, 1]$ and can be realized by a small ReLU network with norm bounded by $1 \cdot \sqrt{1+1+1+1}\sqrt{1+1+1} = 2\sqrt{3}$. That is, $\psi \in \mathcal{NN}(2, 2, 2\sqrt{3})$. For each multi-index $\mathbf{n} = (n_1, \ldots, n_d) \in \{0, 1, \ldots, N\}^d$, we define the basis functions by scaling and shifting ψ as

$$\psi_{\mathbf{n}}(\mathbf{x}) := \prod_{i=1}^{d} \psi(Nx_i - n_i), \quad \mathbf{x} \in [0, 1]^d.$$

From this construction, for each fixed n_i, the factor $\psi(Nx_i - n_i)$ is nonzero only when $x_i \in \left[\frac{n_i - 1}{N}, \frac{n_i + 1}{N}\right]$. Hence, each $\psi_{\mathbf{n}}$ has compact support localized around the grid point $\mathbf{n}/N$, and for any $\mathbf{x} \in [0, 1]^d$, at most 2^d such functions are active. Moreover, these functions form a partition of unity:

$$\sum_{\mathbf{n}} \psi_{\mathbf{n}}(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in [0, 1]^d,$$

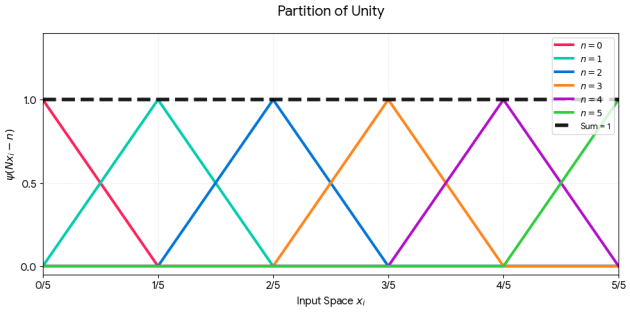

Figure 3: Partition of Unity via Localized Hat Functions. The plot illustrates the basis functions $\psi(Nx_i - n_i)$ for $N = 5$ and indices $n_i \in \{0, \ldots, 5\}$. Each colored triangle represents a localized basis function centered at $x_i = n_i/N$ with support on $[\frac{n_i-1}{N}, \frac{n_i+1}{N}]$. The dashed black line indicates the sum $\sum_n \psi_n(x)$, which is identically equal to 1 across the domain $[0, 1]$.

which follows from the corresponding one-dimensional property of ψ and the tensor-product structure. This locality leads to representations that depend only on nearby regions of the domain, which is crucial for stable approximation and error control. Figure 3 illustrates this construction.

For any $h \in \mathcal{C}^\alpha([0,1]^d)$, set its local Taylor expansion

$$p(\mathbf{x}) = \sum_{\mathbf{n}} \sum_{\|\mathbf{s}\|_1 \le r} c_{\mathbf{n},\mathbf{s}} p_{\mathbf{n},\mathbf{s}}(\mathbf{x}), \quad p_{\mathbf{n},\mathbf{s}}(x) = \psi_{\mathbf{n}}(\mathbf{x}) \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}},$$

where $c_{\mathbf{n},\mathbf{s}} = \dfrac{\partial^{\mathbf{s}} h\left(\frac{\mathbf{n}}{N}\right)}{\mathbf{s}!} \le \mathbf{1}$ and $p_{\mathbf{n},\mathbf{s}}$ is supported on $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \frac{\mathbf{n}}{N}\|_\infty \le \frac{1}{N}\}$. Then

$$\begin{aligned}
|h(\mathbf{x}) - p(\mathbf{x})| &\le \sum_{\mathbf{n}} \psi_{\mathbf{n}}(\mathbf{x}) \left| h(\mathbf{x}) - \sum_{\|\mathbf{s}\|_1 \le r} c_{\mathbf{n},\mathbf{s}} \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}} \right| \\
&\le \sum_{\mathbf{n}: \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty \le \frac{1}{N}} \psi_{\mathbf{n}}(\mathbf{x}) \left| h(\mathbf{x}) - \sum_{\|\mathbf{s}\|_1 \le r} c_{\mathbf{n},\mathbf{s}} \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}} \right| \\
&\le \sum_{\mathbf{n}: \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty \le \frac{1}{N}} d^r \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty^\alpha \le 2^d d^r N^{-\alpha},
\end{aligned}$$

where we used the Taylor remainder estimate with $\boldsymbol{\xi}$ lying on the line segment between $\mathbf{x}$ and $\frac{\mathbf{n}}{N}$g

$$\begin{aligned}
\left| h(\mathbf{x}) - \sum_{\|\mathbf{s}\|_1 \le r} c_{\mathbf{n},\mathbf{s}} \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}} \right| &= \left| \sum_{\|\mathbf{s}\|_1 = r} \frac{1}{\mathbf{s}!} \left(\partial^{\mathbf{s}} h(\boldsymbol{\xi}) - \partial^{\mathbf{s}} h\left(\frac{\mathbf{n}}{N}\right)\right) \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}} \right| \\
&\le \sum_{\|\mathbf{s}\|_1 = r} \frac{1}{\mathbf{s}!} \left| \partial^{\mathbf{s}} h(\boldsymbol{\xi}) - \partial^{\mathbf{s}} h\left(\frac{\mathbf{n}}{N}\right) \right| \cdot \left| \left(\mathbf{x} - \frac{\mathbf{n}}{N}\right)^{\mathbf{s}} \right| \\
&\le \sum_{\|\mathbf{s}\|_1 = r} \frac{1}{\mathbf{s}!} \left\|\boldsymbol{\xi} - \frac{\mathbf{n}}{N}\right\|_\infty^{\alpha - r} \cdot \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty^r \\
&\le \left(\sum_{\|\mathbf{s}\|_1 = r} \frac{1}{\mathbf{s}!} \right) \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty^\alpha \le d^r \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty^\alpha
\end{aligned}$$

and at most 2^d functions $\psi_{\mathbf{n}}$ overlap at any point $\mathbf{x}$ for the last step.

Step 2: Approximate $p(\mathbf{x})$ by neural networks. Note that there are $d + \|\mathbf{s}\|_1 \le t$ multiplier factors in $p_{\mathbf{n},\mathbf{s}}(\mathbf{x})$, where $t = d + r$. By Lemma 3, for any $k \in \mathbb{N}$, there exists a network $\phi_t \in$

$\mathcal{NN}(6tk, 2\lceil \log_2 t \rceil, K_t)$ that approximates the t-fold product on $[-1,1]^t$ with error

$$\left| \prod_{i=1}^{t} x_i - \phi_t(x_1, \ldots, x_t) \right| \leq \frac{6t}{k^2}, \tag{11}$$

and its norm K_t satisfies

$$K_t \leq 722^{\lceil \log_2 t \rceil} \cdot 2^{\frac{7\lceil \log_2 t \rceil^2 - 7\lceil \log_2 t \rceil}{4}}. \tag{12}$$

Next, we construct a network for each $p_{\mathbf{n},\mathbf{s}}$ and then do the linear combination of them. Slightly abusing the notation, we observe that

$$\psi(Nx_i - n_i) = \sigma(1 - \sigma(Nx_i - n_i) - \sigma(-Nx_i + n_i))$$

and $1 \cdot \sqrt{1+1+1+1} \cdot \sqrt{2N^2 + 2n_i^2 + 1} \leq 2\sqrt{5}N \leq 2\sqrt{15}N$, therefore $\psi(Nx_i - n_i) \in \mathcal{NN}(2, 2, 2\sqrt{15}N)$.

In addition, it is straightforward to observe that

$$\begin{aligned}\left(x_i - \frac{n_i}{N}\right) &= \sigma\left(x_i - \frac{n_i}{N}\right) - \sigma\left(-x_i + \frac{n_i}{N}\right) \\ &= \sigma\left(\sigma\left(x_i - \frac{n_i}{N}\right)\right) - \sigma\left(\sigma\left(-x_i + \frac{n_i}{N}\right)\right)\end{aligned}$$

and thus $2\sqrt{1+1+1} \cdot \sqrt{1+1+2\frac{n_i^2}{N^2}+1} \leq 2\sqrt{15} \leq 2\sqrt{15}N$. This implies $\left(x_i - \frac{n_i}{N}\right) \in \mathcal{NN}(2, 2, 2\sqrt{3}\sqrt{5})$. By Proposition 2.2, it follows that

$$\left(\psi(Nx_1 - n_1), \ldots, \psi(Nx_d - n_d), \ldots, x_i - \frac{n_i}{N}, \ldots\right) \in \mathcal{NN}(2(d+r), 2, 2\sqrt{15}(d+r)\sqrt{d+r}N).$$

Then, by (11) and by Proposition 2.3 we obtain a network

$$\phi_{\mathbf{n},\mathbf{s}} = \phi_t(\psi(Nx_1 - n_1), \ldots, \psi(Nx_d - n_d), \ldots, x_i - n_i, \ldots)$$

such that $W_{\phi_{\mathbf{n},\mathbf{s}}} = 12(d+r)k$, $D_{\phi_{\mathbf{n},\mathbf{s}}} = 2\lceil \log_2(d+r) \rceil + 2$ and

$$|\phi_{\mathbf{n},\mathbf{s}}(\mathbf{x}) - p_{\mathbf{n},\mathbf{s}}(\mathbf{x})| \leq \frac{6(d+r)}{k^2}.$$

For its norm bound, since $2\sqrt{15}(d+r+1)\sqrt{d+r}N)K_{\phi_t} \leq 8(d+r)^2 N K_{\phi_t}$, we have

$$K_{\phi_{\mathbf{n},\mathbf{s}}} \leq 8(d+r)^2 N \cdot 722^{\lceil \log_2(2(d+r)) \rceil} \cdot 2^{\frac{7\lceil \log_2(2(d+r)) \rceil^2 - 7\lceil \log_2(2(d+r)) \rceil}{4}}.$$

Moreover, since $\phi_{\mathbf{n},\mathbf{s}}(x_1, \ldots, x_t) = 0$ when $x_1 \cdot x_2 \cdots x_t = 0$, $\phi_{\mathbf{n},\mathbf{s}}$ is supported on $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \frac{\mathbf{n}}{N}\|_\infty \leq \frac{1}{N}\}$.

Define the following linear combination accordingly to approximate $p(\mathbf{x})$

$$\phi(\mathbf{x}) = \sum_{\mathbf{n}} \sum_{\|\mathbf{s}\|_1 \leq r} c_{\mathbf{n},\mathbf{s}} \phi_{\mathbf{n},\mathbf{s}}(\mathbf{x}).$$

There are less than $(N+1)^d d^r$ terms in the summation. By the linear combination (Proposition 2.1) and (12), we have

$$\phi(\mathbf{x}) \in \mathcal{NN}(12(N+1)^d d^r (d+r)k, 2\lceil \log_2(d+r) \rceil + 2, K_\phi)$$

and the norm constraint K_ϕ can be bounded, said $D_\phi = 2\lceil \log_2(d+r) \rceil + 2$, by

$$K_\phi \leq \left((N+1)^d d^r + 1\right)^{\frac{D_\phi + 1}{2}} 8(d+r)^2 N \cdot 722^{\lceil \log_2(2(d+r)) \rceil} \cdot 2^{\frac{7\lceil \log_2(2(d+r)) \rceil^2 - 7\lceil \log_2(2(d+r)) \rceil}{4}}.$$

The approximation error

$$\begin{aligned}|p(\mathbf{x}) - \phi(\mathbf{x})| &= \left| \sum_{\mathbf{n}} \sum_{\|\mathbf{s}\|_1 \leq r} c_{\mathbf{n},\mathbf{s}} p_{\mathbf{n},\mathbf{s}}(\mathbf{x}) - \sum_{\mathbf{n}} \sum_{\|\mathbf{s}\|_1 \leq r} c_{\mathbf{n},\mathbf{s}} \phi_{\mathbf{n},\mathbf{s}}(\mathbf{x}) \right| \\ &\leq \sum_{\mathbf{n}: \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty \leq \frac{1}{N}} \sum_{\|\mathbf{s}\|_1 \leq r} |c_{\mathbf{n},\mathbf{s}}| \cdot |p_{\mathbf{n},\mathbf{s}}(\mathbf{x}) - \phi_{\mathbf{n},\mathbf{s}}(\mathbf{x})| \\ &\leq \sum_{\mathbf{n}: \left\|\mathbf{x} - \frac{\mathbf{n}}{N}\right\|_\infty \leq \frac{1}{N}} \sum_{\|\mathbf{s}\|_1 \leq r} 1 \cdot \frac{6(d+r)}{k^2} \\ &= 2^d d^r \cdot \frac{6(d+r)}{k^2}\end{aligned}$$

where we used that 2^d functions $\psi_{\mathbf{n}}$ overlap at any point $\mathbf{x}$ for the last step

Step 3: Combining error bounds. The overall approximation error is bounded by

$$\begin{aligned}
&\|h-\phi\|_{L^\infty([0,1]^d)} \\
&\le \|h-p\|_{L^\infty([0,1]^d)} + \|p-\phi\|_{L^\infty([0,1]^d)} \\
&\le 2^d d^r N^{-\alpha} + \frac{6\cdot 2^d(d+r)d^r}{k^2}.
\end{aligned} \tag{13}$$

It gives a rate of order k^{-2} by choosing $N \asymp \lceil k^{2/\alpha}\rceil$. Then the associated network has width $W \asymp k^{2d/\alpha+1}$, depth $D = 2\lceil \log_2(d+r)\rceil + 2$ and norm constraint $K \asymp N^{d(D+1)/2+1} \asymp k^{(d(D+1)+2)/\alpha}$. Since the approximation rate in (13) decreases with increasing N and k (increasing width W) for a fixed depth, it follows that

$$\|h-\phi\|_{L^\infty([0,1]^d)} \lesssim K^{-\frac{2\alpha}{2+d(D+1)}}$$

provided that $W \gtrsim K^{\frac{2d+\alpha}{d(D+1)+2}}$. This completes the proof. □

B.2 Proof of approximating sparse compositional functions

Before presenting the proof of Theorem 2, we show that the result extends to intermediate functions of the form

$$g_{v^{(\ell)}} \in C^{\alpha_{v^{(\ell)}}}\Big([-R_{v^{(\ell-1)}}, R_{v^{(\ell-1)}}]^{d_{\text{in}(v^{(\ell)})}}\Big), \quad \|g_{v^{(\ell)}}\|_\infty \le R_{v^{(\ell)}},$$

since they can be reduced to the unit-cube setting via a simple rescaling. For $\ell = 1,\dots,L-1$, define

$$h_{\mathcal{V}^{(\ell)}}(x) := \frac{g_{\mathcal{V}^{(\ell)}}\big(2R_{\mathcal{V}^{(\ell-1)}}x - R_{\mathcal{V}^{(\ell-1)}}\big)}{2R_{\mathcal{V}^{(\ell)}}} + \frac{1}{2},$$

where the affine transformation $2R_{\mathcal{V}^{(\ell-1)}}x - R_{\mathcal{V}^{(\ell-1)}}$ is applied coordinate-wise, i.e., each parent variable x_u is mapped to $2R_u x_u - R_u$. At the boundary levels, define

$$h_{\mathcal{V}^{(0)}} := \frac{g_{\mathcal{V}^{(0)}}}{2R_0} + \frac{1}{2}, \qquad h_{\mathcal{V}^{(L)}}(x) := g_{\mathcal{V}^{(L)}}\big(2R_{\mathcal{V}^{(L-1)}}x - R_{\mathcal{V}^{(L-1)}}\big).$$

Then, for $\ell = 1,\dots,L-1$, we have

$$h_{v^{(\ell)}} \in C^{\alpha_{v^{(\ell)}}}\big([0,1]^{d_{\text{in}(v^{(\ell)})}}, 1\big), \quad \text{and} \quad h_{v^{(L)}} \in C^{\alpha_{v^{(L)}}}\big([0,1]^{|\text{pa}(v^{(L)})|}, R_{v^{(L)}}\big).$$

Consequently, the composition f can be equivalently written as

$$f = g_{\mathcal{V}^{(L)}} \circ \cdots \circ g_{\mathcal{V}^{(0)}} = h_{\mathcal{V}^{(L)}} \circ \cdots \circ h_{\mathcal{V}^{(0)}}.$$

Proof of Theorem 2. The proof contains three parts.

(Networks construction). By Theorem 1, each $g_{v^{(\ell)}}$ can be approximated by a neural network $\phi_{v^{(\ell)}} \in \mathcal{NN}(W_{v^{(\ell)}}, D_{v^{(\ell)}}, K_{v^{(\ell)}})$ with

$$\begin{aligned}
W_{v^{(\ell)}} &= 12(d_{\text{in}(v^{(\ell)})} + r_{v^{(\ell)}}) d_{\text{in}(v^{(\ell)})}{}^{r_{v^{(\ell)}}} (N_{v^{(\ell)}}+1)^{d_{\text{in}(v^{(\ell)})}} k_{v^{(\ell)}}, \\
D_{v^{(\ell)}} &= 2\left\lceil \log_2(d_{\text{in}(v^{(\ell)})} + r_{v^{(\ell)}})\right\rceil + 2, \\
K_{v^{(\ell)}} &= \Big((N_{v^{(\ell)}}+1)^{d_{\text{in}(v^{(\ell)})}} d_{\text{in}(v^{(\ell)})}^{r_{v^{(\ell)}}} + 1\Big)^{\frac{D_{v^{(\ell)}}+1}{2}} 8(d_{\text{in}(v^{(\ell)})} + r_{v^{(\ell)}})^2 N_{v^{(\ell)}} \\
&\quad \cdot 722^{\lceil \log_2(2(d_{\text{in}(v^{(\ell)})}+r_{v^{(\ell)}}))\rceil} \cdot 2^{\frac{7\lceil \log_2(2(d_{\text{in}(v^{(\ell)})}+r_{v^{(\ell)}}))\rceil^2 - 7\lceil \log_2(2(d_{\text{in}(v^{(\ell)})}+r_{v^{(\ell)}}))\rceil}{4}}.
\end{aligned} \tag{14}$$

and

$$\begin{aligned}
&\|g_{v^{(\ell)}} - \phi_{v^{(\ell)}}\|_{L^\infty([0,1]^{d_{\text{in}(v^{(\ell)})}})} \\
&\le 2^{d_{\text{in}(v^{(\ell)})}} d_{\text{in}(v^{(\ell)})}{}^{r_{v^{(\ell)}}} \left(N_{v^{(\ell)}}^{-\alpha_{v^{(\ell)}}} + \frac{6(d_{\text{in}(v^{(\ell)})} + r_{v^{(\ell)}}) d_{\text{in}(v^{(\ell)})}{}^{r_{v^{(\ell)}}}}{k_{v^{(\ell)}}^2}\right)
\end{aligned} \tag{15}$$

where the precise constants are given by the proof of Theorem 1. By the concatenation property (Proposition 2.2) of neural networks, define

$$\phi_{\mathcal{V}^{(\ell)}} := \left(\phi_{v_1^{(\ell)}}, \ldots, \phi_{v_{|\mathcal{V}^{(\ell)}|}^{(\ell)}}\right) \in \mathcal{NN}\left(W_{\mathcal{V}^{(\ell)}}, D_{\mathcal{V}^{(\ell)}}, K_{\mathcal{V}^{(\ell)}}\right).$$

Here, the width $W_{\mathcal{V}^{(\ell)}} = \sum_{v^{(\ell)} \in \mathcal{V}^{(\ell)}} W_{v^{(\ell)}}$, the depth $D_{\mathcal{V}^{(\ell)}} = \max_{v^{(\ell)} \in \mathcal{V}^{(\ell)}} \{D_{v^{(\ell)}}\}$, and the norm bound $K_{\mathcal{V}^{(\ell)}} = (|\mathcal{V}^{(\ell)}| + 1)^{(D_{\mathcal{V}^{(\ell)}}+1)/2} \max_{v^{(\ell)} \in \mathcal{V}^{(\ell)}} K_{v^{(\ell)}}$. We then construct the overall network

$$\phi = \phi_{\mathcal{V}^{(L)}} \circ \cdots \circ \phi_{\mathcal{V}^{(1)}},$$

whose width is $W = \max_\ell \{W_{\mathcal{V}^{(\ell)}}\}$, depth is $D = \sum_{\ell=1}^{L} D_{\mathcal{V}^{(\ell)}}$, and norm bound satisfies $K \le 2^{D/2} \prod_\ell (K_{\mathcal{V}^{(\ell)}} + 2)$ by Proposition 2. That is, the resulting network satisfies

$$\phi \in \mathcal{NN}\left(\max_\ell \{W_{\mathcal{V}^{(\ell)}}\}, \sum_{\ell=1}^{L} D_{\mathcal{V}^{(\ell)}}, 2^{D/2} \prod_\ell (K_{\mathcal{V}^{(\ell)}} + 2)\right).$$

(Approximation rate). Denote $G_\ell = g_{\mathcal{V}^{(\ell)}} \circ g_{\mathcal{V}^{(\ell-1)}} \circ \cdots g_{\mathcal{V}^{(0)}}$ and $\Phi_\ell = \phi_{\mathcal{V}_{(\ell)}} \circ \phi_{\mathcal{V}_{(\ell-1)}} \circ \cdots \circ \phi_{\mathcal{V}_{(0)}}$. Then

$$\begin{aligned}
&|f(\mathbf{x}) - \phi(\mathbf{x})| = |G_L(\mathbf{x}) - \Phi_L(\mathbf{x})| \\
&= |g_{\mathcal{V}^{(L)}} \circ \cdots \circ g_{\mathcal{V}^{(0)}}(\mathbf{x}) - \phi_{\mathcal{V}^{(L)}} \circ \cdots \circ \phi_{\mathcal{V}^{(0)}}(\mathbf{x})| \\
&= |g_{\mathcal{V}^{(L)}} \circ G_{L-1}(\mathbf{x}) - g_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x})| + |g_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x}) - \phi_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x})| \\
&= |g_{\mathcal{V}^{(L)}} \circ G_{L-1}(\mathbf{x}) - g_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x})| + |g_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x}) - \phi_{\mathcal{V}^{(L)}} \circ \Phi_{L-1}(\mathbf{x})| \\
&\le |G_{L-1}(\mathbf{x}) - \Phi_{L-1}(\mathbf{x})|_\infty^{\alpha_{v^{(L)}} \wedge 1} + \|g_{\mathcal{V}^{(L)}} - \phi_{\mathcal{V}_{(L)}}\|_{L_\infty([0,1]^{\mathrm{pa}(\mathcal{V}^{(L)})})}.
\end{aligned}$$

Here, $|G(\mathbf{x})|_\infty = \max_i(|G(\mathbf{x})|)_i$ for any vector-valued function $G(\mathbf{x})$. In the last step, we use the fact that $|f(x_1) - f(x_2)| \le \|f\|_{C^\alpha} \|x_1 - x_2\|_\infty^\alpha$ for $0 < \alpha \le 1$, and $|f(x_1) - f(x_2)| \le \|\nabla f\|_\infty \|x_1 - x_2\|_\infty$ for $\alpha > 1$, applied to the first term.

For any $\ell = 1, \ldots, L-1$, we derive the following error iteration

$$\begin{aligned}
&|G_\ell(\mathbf{x}) - \Phi_\ell(\mathbf{x})|_\infty \\
&= \max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})} \Big\{ \left|g_{v^{(\ell)}} \circ G_{\mathrm{pa}(v^{(\ell)})}(\mathbf{x}) - g_{v^{(\ell)}} \circ \Phi_{\mathrm{pa}(v^{(\ell)})}(\mathbf{x})\right| \\
&\qquad\qquad + \left|g_{v^{(\ell)}} \circ \Phi_{\mathrm{pa}(v^{(\ell)})}(\mathbf{x}) - \phi_{v^{(\ell)}} \circ \Phi_{\mathrm{pa}(v^{(\ell)})}(\mathbf{x})\right| \Big\} \\
&\le \max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})} \left\{ |G_{\ell-1}(\mathbf{x}) - \Phi_{\ell-1}(\mathbf{x})|_\infty^{\alpha_{v^{(\ell)}} \wedge 1} + \|g_{v^{(\ell)}} - \phi_{v^{(\ell)}}\|_{L_\infty([0,1]^{d_{\mathrm{in}(v^{(\ell)})}})} \right\}. \qquad (16)
\end{aligned}$$

Recalling $\mathrm{pa}(v^{(\ell)}) = \{v^{(\ell-1)} : (v^{(\ell-1)}, v^{(\ell)}) \in \mathcal{E}^{(\ell)}\}$, we write $G_{\mathrm{pa}(v^{(\ell)})}$ and $\Phi_{\mathrm{pa}(v^{(\ell)})}$ instead of $G_{\ell-1}$ and $\Phi_{\ell-1}$ to emphasize the dependency along the path. To interpret $\max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})}$ appearing above, it is necessary to specify which node $v^{(\ell+1)} \in \mathcal{V}^{(\ell+1)}$ is selected. This choice is determined by the error propagation from the output layer backward. Since there is a single output node at level L, the critical path is identified by selecting, at each level, the node that maximizes the propagated error. For instance, we define $v_*^{(L-1)} := \arg\max_i \left(|G_{L-1}(\mathbf{x}) - \Phi_{L-1}(\mathbf{x})|\right)_i$, and subsequently select $v^{(L-2)} \in \mathrm{pa}(v_*^{(L-1)})$, and so on. In this sense, the specific node $v^{(\ell+1)}$ is implicitly determined by the error values established in the preceding levels of the recursion and $\max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})}$ is actually $\max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})} \cdots \max_{v^{(L-1)} \in \mathrm{pa}(v^{(L)})}(\cdot)$. To simplify notation, let $\pi_{\ell \to L} := v^{(\ell)} \to v^{(\ell+1)} \to \ldots \to v^{(L)}$ denote the path selected by the above recursion. We then write

$$\max_{v^{(\ell)} \in \mathrm{pa}(v^{(\ell+1)})} \cdots \max_{v^{(L-1)} \in \mathrm{pa}(v^{(L)})}(\cdot) = \max_{\pi_{\ell \to L}}(\cdot).$$

Denote

$$A_\ell = |G_\ell(\mathbf{x}) - \Phi_\ell(\mathbf{x})|_\infty, \qquad B_\ell = \|g_{v^{(\ell)}} - \phi_{v^{(\ell)}}\|_{L_\infty([0,1]^{d_{\mathrm{in}(v^{(\ell)})}})}, \qquad \tilde{\alpha}_\ell = \min\{\alpha_{v^{(\ell)}}, 1\}.$$

Then, from (16),

$$A_L = |G_L(\mathbf{x}) - \Phi_L(\mathbf{x})|, \qquad A_\ell \le \max_{\pi_{\ell\to L}} \sqrt{d} A_{\ell-1}^{\tilde{\alpha}_\ell} + B_\ell, \qquad A_0 = 0.$$

Using $(x+y)^\alpha \le x^\alpha + y^\alpha$ for $\alpha \in (0,1]$ and iterating the recursion, we obtain

$$\begin{aligned} A_L &\le \max_{\pi_{1\to L}} \left\{ B_1^{\tilde{\alpha}_2\tilde{\alpha}_3\cdots\tilde{\alpha}_L} + d^{(L-2)/2} B_2^{\tilde{\alpha}_3\cdots\tilde{\alpha}_L} + \cdots + \sqrt{d} B_{L-1}^{\tilde{\alpha}_L} + B_L \right\} \\ &\le L \max_{v^{(\ell)}:\pi_{1\to L}} B_\ell^{\prod_{j=\ell+1}^L \tilde{\alpha}_j}. \end{aligned} \tag{17}$$

To clarify, the notation $\max_{v^{(\ell)}:\pi_{1\to L}}$ refers to selecting the node $v^{(\ell)}$ along the path $\pi_{1\to L}$ that maximizes the quantity $B_\ell^{\prod_{j=\ell+1}^L \tilde{\alpha}_j}$. The exponent $\prod_{j=\ell+1}^L \tilde{\alpha}_j$ represents the cumulative regularity along the subpath $v^{(\ell)} \to \cdots \to v^{(L)}$, which is part of the full path $v^{(1)} \to \cdots \to v^{(L)}$. In other words, error propagation follows a forward evaluation, while the critical path is identified by backward tracing of the maximal error from the output node. Therefore,

$$|f(x) - \phi(x)| \le L \max_{v^{(\ell)}:\pi_{1\to L}} \|g_{v^{(\ell)}} - \phi_{v^{(\ell)}}\|_{L_\infty([0,1]^{d_{\text{in}(v^{(\ell)})}})}^{\prod_{j=\ell+1}^L \tilde{\alpha}_j}. \tag{18}$$

(Parameter tuning). By (14) and (15), taking $N_{v^{(\ell)}} \asymp \left\lceil k_{v^{(\ell)}}^{2/\alpha_{v^{(\ell)}}} \right\rceil$ gives the width for each node $W_{v^{(\ell)}} \asymp k_{v^{(\ell)}}^{(2d_{\text{in}(v^{(\ell)})}+\alpha_{v^{(\ell)}})/\alpha_{v^{(\ell)}}}$ and the corresponding norm constraint $K_{v^{(\ell)}} \asymp k_{v^{(\ell)}}^{\left(d_{\text{in}(v^{(\ell)})}(D_{v^{(\ell)}}+1)+2\right)/\alpha_{v^{(\ell)}}}$. Then, the parameters for the overall network ϕ become

$$W = \max_\ell \left\{ \sum_{v^{(\ell)}} W_{v^{(\ell)}} \right\} \asymp \max_{\ell, v^{(\ell)}} k_{v^{(\ell)}}^{\frac{2d_{\text{in}(v^{(\ell)})}+\alpha_{v^{(\ell)}}}{\alpha_{v^{(\ell)}}}} \asymp \max_{\ell, v^{(\ell)}} K_{v^{(\ell)}}^{\frac{2d_{\text{in}(v^{(\ell)})}+\alpha_{v^{(\ell)}}}{2+d_{\text{in}(v^{(\ell)})}\left(D_{v^{(\ell)}}+1\right)}}$$

$$D = \sum_{\ell=1}^L \max_{v^{(\ell)}\in\mathcal{V}^{(\ell)}} D_{v^{(\ell)}} = 2L \max_{\ell, v^{(\ell)}} \left\lceil \log_2(d_{\text{in}(v^{(\ell)})} + r_{v^{(\ell)}}) \right\rceil + 2L$$

$$K = 2^{D/2} \prod_{\ell=1}^L \left(\left(|\mathcal{V}^{(\ell)}| + 1 \right)^{\frac{D_{\mathcal{V}^{(\ell)}}+1}{2}} \max_{v^{(\ell)}\in\mathcal{V}^{(\ell)}} K_{v^{(\ell)}} + 2 \right) \asymp \left(\max_{\ell, v^{(\ell)}} K_{v^{(\ell)}} \right)^L.$$

Therefore, for any $W \gtrsim \max_{\ell, v^{(\ell)}} K^{\frac{2d_{\text{in}(v^{(\ell)})}+\alpha_{v^{(\ell)}}}{d_{\text{in}(v^{(\ell)})}\left(D_{v^{(\ell)}}+1\right)L+2L}}$, combining (15) and (18), we obtain

$$\begin{aligned} \|f - \phi\|_\infty &\le L \max_{v^{(\ell)}:\pi_{1\to L}} K_{v^{(\ell)}}^{-\frac{2\alpha_{v^{(\ell)}}}{2+d_{\text{in}(v^{(\ell)})}\left(D_{v^{(\ell)}}+1\right)} \prod_{m=\ell+1}^L \tilde{\alpha}_{v^{(m)}}} \\ &\le L \max_{v^{(\ell)}:\pi_{1\to L}} K^{-\frac{2\alpha^*_{v^{(\ell)}}}{2L+(D+L)d_{\text{in}(v^{(\ell)})}}} \end{aligned}$$

where $\alpha^*_{v^{(\ell)}} := \alpha_{v^{(\ell)}} \prod_{m=\ell+1}^L \{\alpha_{v^{(m)}}, 1\}$. This completes the proof. □

C Proof of the statistical analysis of Frobenius norm-constrained neural networks

In this section, we combine the approximation error established in Theorem 2 with estimation error bounds to derive the excess risk.

As discussed in Section 4, the empirical Rademacher complexity is a standard tool for controlling the estimation error. Let $\mathcal{G}$ be a class of real-valued functions defined on $\mathcal{X}$. The empirical Rademacher complexity of $\mathcal{G}$ with respect to a sample $S = \{x_1, \ldots, x_n\}$ is defined as

$$\hat{\mathfrak{R}}_n(\mathcal{G}) := \mathbb{E}_\sigma \left[\sup_{g\in\mathcal{G}} \frac{1}{n} \sum_{i=1}^n \sigma_i g(x_i) \right],$$

where $\sigma_1, \ldots, \sigma_n$ are independent Rademacher variables, i.e., random variables uniformly distributed over $\{-1, +1\}$. Prior work [63, 11, 35] provide Rademacher complexity bounds for function classes generated by neural networks of the form

$$\phi(x) = \widetilde{A}_D \sigma(\widetilde{A}_{D-1} \sigma(\cdots \sigma(\widetilde{A}_0 \tilde{x}))). \tag{19}$$

Here, the bias is absorbed into the augmented matrices $\widetilde{A}_\ell$, where

$$\widetilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad \widetilde{A}_D = (A_D, 0), \quad \widetilde{A}_\ell = \begin{pmatrix} A_\ell & b_\ell \\ \mathbf{0} & 1 \end{pmatrix}, \quad \ell = 0, \ldots, D-1,$$

with $\widetilde{A}_D \in \mathbb{R}^{N_{D+1} \times (N_D+1)}$ and $\widetilde{A}_\ell \in \mathbb{R}^{(N_{\ell+1}+1) \times (N_\ell+1)}$. Analogously to the function class $\mathcal{NN}(W, D)$ including (1), we denote this function class by $\widetilde{\mathcal{NN}}(W, D)$. The corresponding Frobenius norm constraint is defined as

$$\tilde{\kappa}(\theta) = \prod_{\ell=0}^{D} \|\tilde{A}_\ell\|_F = \|A_D\|_F \prod_{\ell=0}^{D-1} \left(\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1\right)^{1/2} = \kappa(\theta). \tag{20}$$

Analogously to $\mathcal{NN}(W, D, K)$ used in the approximation analysis, we define the constrained class

$$\widetilde{\mathcal{NN}}(W, D, K) = \{f_\theta \in \widetilde{\mathcal{NN}}(W, D) : \tilde{\kappa}(\theta) \leq K\}.$$

Before presenting the proof of Theorem 3, the following lemma summarizes the relation between the two neural network classes.

Lemma 4. *For any W, D, and $K \geq 0$, we have*

$$\widetilde{\mathcal{NN}}(W, D, K) \equiv \mathcal{NN}(W, D, K).$$

Moreover, $\tilde{k}(\theta) = k(\theta)$ for any choice of parameters θ.

The above lemma shows that the two neural network classes can essentially be regarded as equivalent, with a slight abuse of notation in identifying x with the augmented variable $\tilde{x}$. This allows us to leverage results from prior work in our analysis. For completeness, we provide the proof below.

Proof of Lemma 4. We prove the two inclusions separately.

First, consider any function $\tilde{f}_\theta \in \widetilde{\mathcal{NN}}(W, D, K)$. Fixed $\tilde{x}$, expanding the augmented matrix multiplication in $\tilde{f}_\theta(\tilde{x})$ (19), we recover the standard biased form (1) of $f_\theta(x)$. Moreover, by the definition of the augmented norm, the corresponding parameters satisfy $\kappa(\theta) = \widetilde{\kappa}(\tilde{\theta})$. Hence the function $\tilde{f}_\theta$ belongs to $\mathcal{NN}(W, D, K)$.

Conversely, let $f_\theta \in \mathcal{NN}(W, D, K)$ be represented in the standard form (1). Define the augmented input $\widetilde{x} = (x^\top, 1)^\top$ and the augmented matrices

$$\widetilde{A}_D = (A_D, 0), \qquad \widetilde{A}_\ell = \begin{pmatrix} A_\ell & b_\ell \\ \mathbf{0} & 1 \end{pmatrix}, \qquad \ell = 0, \ldots, D-1.$$

Then

$$f_\theta(x) = \widetilde{A}_D \sigma\big(\widetilde{A}_{D-1} \sigma(\cdots \sigma(\widetilde{A}_0 \widetilde{x}))\big).$$

Therefore $f_\theta \in \widetilde{\mathcal{NN}}(W, D, K)$. Furthermore,

$$\widetilde{\kappa}(\theta) = \|\widetilde{A}_D\|_F \prod_{\ell=0}^{D-1} \|\widetilde{A}_\ell\|_F = \|A_D\|_F \prod_{\ell=0}^{D-1} \left(\|A_\ell\|_F^2 + \|b_\ell\|_2^2 + 1\right)^{1/2} = \kappa(\theta).$$

This completes the proof. □

The proof of Theorem 3 is now given.

Proof of Theorem 3. **(Network construction)** To obtain bounded outputs, we apply the truncation operator $\chi_1(x) = \max\{-1, \min\{x, 1\}\} \in [-1, 1]$ to functions in $\mathcal{NN}(W, D, K)$. This operator admits a ReLU realization of the form

$$\chi_1(x) = \sigma(x) - \sigma(-x) - \sigma(x-1) + \sigma(-x-1).$$

(Error estimation) With $\tilde{f} := \arg\min_{f \in \mathcal{NN}_1(W,D,K)} \mathcal{R}(f)$, the excess risk can be decomposed as

$$\begin{aligned}
&\mathcal{R}(\hat{f}) - \mathcal{R}(f_*) \\
&= \mathcal{R}(\hat{f}) - \hat{\mathcal{R}}(\hat{f}) + \hat{\mathcal{R}}(\hat{f}) - \hat{\mathcal{R}}(\tilde{f}) + \hat{\mathcal{R}}(\tilde{f}) - \mathcal{R}(\tilde{f}) + \mathcal{R}(\tilde{f}) - \mathcal{R}(f_*) \\
&\leq \mathcal{R}(\hat{f}) - \hat{\mathcal{R}}(\hat{f}) + \hat{\mathcal{R}}(\tilde{f}) - \mathcal{R}(\tilde{f}) + \mathcal{R}(\tilde{f}) - \mathcal{R}(f_*) \\
&\leq \sup_{f \in \mathcal{NN}_1(W,D,K)} \{\mathcal{R}(f) - \hat{\mathcal{R}}(f)\} + \sup_{f \in \mathcal{NN}_1(W,D,K)} \{\hat{\mathcal{R}}(f) - \mathcal{R}(f)\} + \mathcal{R}(\tilde{f}) - \mathcal{R}(f_*). \qquad (21)
\end{aligned}$$

For the approximation error $\mathcal{R}(\tilde{f}) - \mathcal{R}(f_*)$, let $\phi \in \mathcal{NN}(W, D, K)$ and define $f = \chi_1 \circ \phi \in \mathcal{NN}_1(W, D, K)$. Then, for all $x \in X$, $|f(x) - f_*(x)| \leq |\phi(x) - f_*(x)|$, since $f_* \in [-1, 1]$ and χ_1 is 1-Lipschitz. Thus, it follows from Theorem 2 that

$$\begin{aligned}
\mathcal{R}(\tilde{f}) - \mathcal{R}(f_*) &= \inf_{f \in \mathcal{NN}_1(W,D,K)} \|f - f_*\|_2^2 \\
&\leq \inf_{\phi \in \mathcal{NN}(W,D,K)} \|\phi - f_*\|_2^2 \leq \max_{v^{(\ell)}:\pi_{1\to L}} K^{-\frac{4\alpha^*_{v^{(\ell)}}}{2L+(D+L)d_{\text{in}(v^{(\ell)})}}}. \qquad (22)
\end{aligned}$$

We proceed by bounding the first estimation error term in (21). The second term can be bounded similarly. Let $\mathcal{G}$ be a class of real-valued functions defined on $X \times \mathbb{R}$, such as

$$\mathcal{G} := \left\{ g_f \;\middle|\; g_f(x, y) = (f(x) - y)^2, f \in \mathcal{NN}_1(W, D, K) \right\}.$$

For any $f \in \mathcal{NN}_1(W, D, K)$, we have $||g_f||_\infty \leq 4$ since $|y| \leq 1$. Moreover, it is straightforward to observe that

$$\sup_{f \in \mathcal{NN}_1(W,D,K)} \mathcal{R}(f) - \hat{\mathcal{R}}(f) = \sup_{g_f \in \mathcal{G}} \left(\mathbb{E}_\rho[g_f(x, y)] - \frac{1}{n} \sum_{i=1}^n g_f(x_i, y_i) \right).$$

By the standard one-sided Rademacher complexity bound and McDiarmid inequality for uniformly bounded function classes, see for example [59, Theorem 3.3], with probability at least $1 - \delta/2$,

$$\sup_{g_f \in \mathcal{G}} \left\{ \mathbb{E}_\rho[g_f] - \frac{1}{n} \sum_{i=1}^n g_f(x_i, y_i) \right\} \leq 2\hat{\mathfrak{R}}_n(\mathcal{G}) + 12\sqrt{\frac{\log(4/\delta)}{2n}}.$$

To bound the complexity of $\mathcal{G}$, we apply the contraction principle of Ledoux–Talagrand [53, 10], which states that the Rademacher complexity is stable under Lipschitz transformations. Observe that $g_f(x, y)$ is a Lipschitz function of $f(x)$ with Lipschitz constant bounded by 4. Therefore, as χ_1 is Lipschitz as well, we obtain

$$\hat{\mathfrak{R}}_n(\mathcal{G}) \leq 4\,\hat{\mathfrak{R}}_n(\mathcal{NN}_1(W, D, K)) \leq 4\,\hat{\mathfrak{R}}_n(\mathcal{NN}(W, D, K))$$

From Lemma 4 and [35, Theorem 3.1], the Rademacher complexity of norm-constrained neural networks can be bounded by

$$\hat{\mathfrak{R}}_n(\mathcal{NN}(W, D, K)) \leq \hat{\mathfrak{R}}_n(\widetilde{\mathcal{NN}}(W + 1, D, K)) \leq \frac{\left(\sqrt{2\log(2)D} + 1\right) K}{\sqrt{n}}.$$

Therefore, with probability at least $1 - \delta/2$, the first estimation error

$$\sup_{f \in \mathcal{NN}_1(W,D,K)} \mathcal{R}(f) - \hat{\mathcal{R}}(f) \leq 8\frac{(\sqrt{2\log(2)D} + 1)\ K}{\sqrt{n}} + 12\sqrt{\frac{\log(4/\delta)}{2n}}.$$

Combining this with the error decomposition (21) and the above approximation bound (22), with probability at least $1-\delta$, there holds

$$\mathcal{R}(\hat{f})-\mathcal{R}(f_*)\lesssim \frac{K\sqrt{D}}{\sqrt{n}}+\sqrt{\frac{\log(4/\delta)}{2n}}+L\max_{v^{(\ell)}:\pi_{1\to L}} K^{-\frac{4\alpha^*_{v^{(\ell)}}}{2L+(D+L)d_{\text{in}(v^{(\ell)})}}}.$$

By taking

$$K\asymp \max_{v^{(\ell)}:\pi_{1\to L}} (n)^{\frac{1}{2}\frac{2L+(D+L)d_{\text{in}(v^{(\ell)})}}{2L+(D+L)d_{\text{in}(v^{(\ell)})}+4\alpha^*_{v^{(\ell)}}}},$$

the following holds with probability at least $1-\delta$,

$$\mathcal{R}(\hat{f})-\mathcal{R}(f_*)\lesssim \sqrt{\log\frac{4}{\delta}}\max_{v^{(\ell)}:\pi_{1\to L}} n^{-\frac{2\alpha^*_{v^{(\ell)}}}{2L+(D+L)d_{\text{in}(v^{(\ell)})}+4\alpha^*_{v^{(\ell)}}}}.$$

This completes the proof.

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are clearly stated in the abstract and introduction, and are formalized in Theorems 2 and 3, including assumptions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses key assumptions (e.g., bounded outputs, smoothness, DAG structure, norm constraints) and their implications; see Assumptions 1, 2, and discussion after theorems.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions are explicitly stated, and complete proofs are provided in the appendix with proper cross-referencing.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: This paper is theoretical and does not include empirical experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: This work does not involve datasets, code, or empirical evaluation.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: No experimental setup is involved.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: No computational experiments are conducted.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics and does not involve sensitive data, human subjects, or harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational and theoretical, without a direct application pathway leading to specific societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release data, models, or systems that pose risks of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets, code, or models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects or crowdsourcing are involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.