

# Deep Network Approximation for Smooth Functions

Jianfeng Lu <sup>\*</sup>      Zuowei Shen <sup>†</sup>      Haizhao Yang <sup>‡</sup>      Shijun Zhang <sup>§</sup>

## Abstract

This paper establishes optimal approximation error characterization of deep ReLU networks for smooth functions in terms of both width and depth simultaneously. To that end, we first prove that multivariate polynomials can be approximated by deep ReLU networks of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  with an approximation error  $\mathcal{O}(N^{-L})$ . Through local Taylor expansions and their deep ReLU network approximations, we show that deep ReLU networks of width  $\mathcal{O}(N \ln N)$  and depth  $\mathcal{O}(L \ln L)$  can approximate  $f \in C^s([0, 1]^d)$  with a nearly optimal approximation rate  $\mathcal{O}(\|f\|_{C^s([0, 1]^d)} N^{-2s/d} L^{-2s/d})$ . Our estimate is non-asymptotic in the sense that it is valid for arbitrary width and depth specified by  $N \in \mathbb{N}^+$  and  $L \in \mathbb{N}^+$ , respectively.

**Key words.** Deep ReLU Network, Smooth Function, Polynomial Approximation, Function Composition, Curse of Dimensionality.

## 1 Introduction

Deep neural networks have made significant impacts in many fields of computer science and engineering especially for large-scale and high-dimensional learning problems. Well-designed neural network architectures, efficient training algorithms, and high-performance computing technologies have made neural-network-based methods very successful in tremendous real applications. Especially in supervised learning, e.g., image classification and objective detection, the great advantages of neural-network-based methods have been demonstrated over traditional learning methods. Understanding the approximation capacity of deep neural networks has become a key question for revealing the power of deep learning. A large number of experiments in real applications have shown the large capacity of deep network approximation from many empirical points of view, motivating much effort in establishing the theoretical foundation of deep network approximation. One of the fundamental problems is the characterization of the optimal approximation rate of deep neural networks of arbitrary depth and width.

---

<sup>\*</sup>Department of Mathematics, Department of Physics, and Department of Chemistry, Duke University ([jianfeng@math.duke.edu](mailto:jianfeng@math.duke.edu)).

<sup>†</sup>Department of Mathematics, National University of Singapore ([matzuows@nus.edu.sg](mailto:matzuows@nus.edu.sg)).

<sup>‡</sup>Department of Mathematics, Purdue University ([haizhao@purdue.edu](mailto:haizhao@purdue.edu)).

<sup>§</sup>Department of Mathematics, National University of Singapore ([zhangshijun@u.nus.edu](mailto:zhangshijun@u.nus.edu)).

## 1.1 Main result

Previously, the quantitative characterization of the approximation power of deep feed-forward neural networks (FNNs) with ReLU activation functions is provided in [41]. For ReLU FNNs with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$ , the deep network approximation of  $f \in C([0, 1]^d)$  admits an approximation rate  $\mathcal{O}(\omega_f(N^{-2/d}L^{-2/d}))$  in the  $L^p$ -norm for  $p \in [1, \infty]$ , where  $\omega_f(\cdot)$  is the modulus of continuity of  $f$ . In particular, for the class of Hölder continuous functions, the approximation rate is nearly optimal.<sup>①</sup> The next question is whether the smoothness of functions can improve the approximation rate. In this paper, we investigate the deep network approximation of smaller function space, such as the smooth function space  $C^s([0, 1]^d)$ .

In Theorem 1.1 below, we prove by construction that ReLU FNNs with width  $\mathcal{O}(N \ln N)$  and depth  $\mathcal{O}(L \ln L)$  can approximate  $f \in C^s([0, 1]^d)$  with a nearly optimal approximation rate  $\mathcal{O}(\|f\|_{C^s([0, 1]^d)} N^{-2s/d} L^{-2s/d})$ , where the norm  $\|\cdot\|_{C^s([0, 1]^d)}$  is defined as

$$\|f\|_{C^s([0, 1]^d)} := \max \{ \|\partial^\alpha f\|_{L^\infty([0, 1]^d)} : \|\alpha\|_1 \leq s, \alpha \in \mathbb{N}^d \}, \quad \text{for any } f \in C^s([0, 1]^d).$$

**Theorem 1.1.** *Given a smooth function  $f \in C^s([0, 1]^d)$  with  $s \in \mathbb{N}^+$ , for any  $N, L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $C_1(N + 2) \log_2(8N)$  and depth  $C_2(L + 2) \log_2(4L) + 2d$  such that*

$$\|\phi - f\|_{L^\infty([0, 1]^d)} \leq C_3 \|f\|_{C^s([0, 1]^d)} N^{-2s/d} L^{-2s/d},$$

where  $C_1 = 17s^{d+1}3^d d$ ,  $C_2 = 18s^2$ , and  $C_3 = 85(s + 1)^d 8^s$ .

As we can see from Theorem 1.1, the smoothness improves the approximation rate in  $N$  and  $L$ , e.g.,  $s \geq d$  implies  $\mathcal{O}(N^{-2s/d}L^{-2s/d}) \leq \mathcal{O}(N^{-2}L^{-2})$ . However, we would like to remark that the improved approximation rate is at the price of a much larger prefactor larger than  $d^d$  if  $s \geq d$ . The proof of Theorem 1.1 will be presented in Section 2.2 and its tightness will be discussed in Section 2.3. In fact, the logarithmic terms in width and depth in Theorem 1.1 can be further reduced if the approximation rate is weakened. Note that for any integers

$$\tilde{N} \geq 3(1 + 2)C_1 \log_2(8) = 17s^{d+1}3^{d+2}d \quad \text{and} \quad \tilde{L} \geq C_2(1 + 2) \log_2(4) + 2d = 108s^2 + 2d,$$

there exist  $N, L \in \mathbb{N}^+$  such that

$$C_1(N + 2) \log_2(8N) \leq \tilde{N} < C_1((N + 1) + 2) \log_2(8(N + 1))$$

and

$$C_2(L + 2) \log_2(4L) + 2d \leq \tilde{L} < C_2((L + 1) + 2) \log_2(4(L + 1)) + 2d.$$

It follows that

$$N \geq \frac{N+3}{4} \geq \frac{\tilde{N}}{4C_1 \log_2(8N+8)} \geq \frac{\tilde{N}}{68s^{d+1}3^d d \log_2(8\tilde{N}+8)} \quad \text{and} \quad L \geq \frac{L+3}{4} \geq \frac{\tilde{L}-2d}{4C_2 \log_2(4L+4)} \geq \frac{\tilde{L}-2d}{72s^2 \log_2(4\tilde{L}+4)}.$$

Thus, we have an immediate corollary.

---

<sup>①</sup>“nearly optimal” up to a logarithmic factor.

65 **Corollary 1.2.** *Given a function  $f \in C^s([0, 1]^d)$  with  $s \in \mathbb{N}^+$ , for any  $\tilde{N}, \tilde{L} \in \mathbb{N}^+$ , there*  
 66 *exist a function  $\phi$  implemented by a ReLU FNN with width  $\tilde{N}$  and depth  $\tilde{L}$  such that*

$$67 \quad \|f - \phi\|_{L^\infty([0, 1]^d)} \leq \tilde{C}_1 \|f\|_{C^s([0, 1]^d)} \left( \frac{\tilde{N}}{\tilde{C}_2 \log_2(8\tilde{N}+8)} \right)^{-2s/d} \left( \frac{\tilde{L}-2d}{\tilde{C}_3 \log_2(4\tilde{L}+4)} \right)^{-2s/d},$$

68 *for any  $\tilde{N} \geq 17s^{d+1}3^{d+2}d$  and  $\tilde{L} \geq 108s^2 + 2d$ , where  $\tilde{C}_1 = 85(s+1)^{d8s}$ ,  $\tilde{C}_2 = 68s^{d+1}3^d$ , and*  
 69  *$\tilde{C}_3 = 72s^2$ .*

70 Theorem 1.1 and Corollary 1.2 characterize the approximation rate in terms of total  
 71 number of neurons (with an arbitrary distribution in width and depth) and smoothness  
 72 order of the function to be approximated. In other words, for arbitrary width  $\mathcal{O}(N)$  and  
 73 depth  $\mathcal{O}(L)$ , Theorem 1.1 and Corollary 1.2 provide a nearly optimal approximation rate  
 74  $\mathcal{O}((\frac{N}{\ln N})^{-2s/d}(\frac{L}{\ln L})^{-2s/d})$ . The only result in this direction we are aware of in literature is  
 75 Theorem 4.1 of [45]. It shows that ReLU FNNs with width  $2d+10$  and depth  $L$  achieve an  
 76 nearly optimal rate  $\mathcal{O}((\frac{L}{\ln L})^{-2s/d})$  for sufficiently large  $L$  when approximating functions  
 77 in the unit ball of  $C^s([0, 1]^d)$ . This result is essentially a special case of Theorem 1.1 by  
 78 setting  $N = \mathcal{O}(1)$  and  $L$  sufficiently large.

## 79 1.2 Contributions and related work

80 Our key contributions can be summarized as follows.

81 (i) **Upper bound:** We provide a **quantitative** and **non-asymptotic** approximation  
 82 rate  $\mathcal{O}(\|f\|_{C^s} N^{-2s/d} L^{-2s/d})$  when the ReLU network has width  $\mathcal{O}(N \ln N)$  and depth  
 83  $\mathcal{O}(L \ln L)$  for functions in  $C^s([0, 1]^d)$  in Theorem 1.1. The approximation rate as a  
 84 function of width and depth in this paper is more general and useful than the one  
 85 characterized by the number of nonzero parameters denoted as  $W$  in the literature,  
 86 which is an immediate corollary of our theorem as we shall discuss. In particular,  
 87 our results contain approximation error estimates for both wide networks with fixed  
 88 finite depth and deep networks with fixed finite width.

89 (ii) **Lower bound:** Through the VC-dimension upper bound of ReLU FNNs in [22],  
 90 we prove a lower bound

$$91 \quad C(N^2 L^2 (\ln N)^3 (\ln L)^3)^{-s/d}, \quad \text{for some positive constant } C,$$

92 for the approximation rate of the functions in the unit ball of  $C^s([0, 1]^d)$  approxi-  
 93 mated by ReLU FNNs with width  $\mathcal{O}(N \ln N)$  and depth  $\mathcal{O}(L \ln L)$  in Section 2.3.  
 94 Thus, the approximation rate  $\mathcal{O}(N^{-2s/d} L^{-2s/d})$  is nearly optimal for the unit ball  
 95 of  $C^s([0, 1]^d)$ .

96 (iii) **Approximation of polynomials:** It is proved by construction in Proposition 4.1  
 97 that ReLU FNNs with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  can approximate polynomials  
 98 on  $[0, 1]^d$  with an approximation rate  $\mathcal{O}(N^{-L})$ . This is a non-trivial extension of  
 99 the result  $\mathcal{O}(2^{-L})$  for polynomial approximation by very deep ReLU FNNs in [43].

(iv) **Uniform approximation:** The approximation rate in this paper is measured in the  $L^\infty([0,1]^d)$ -norm as a result of Theorem 2.1. To achieve this, given a ReLU FNN  $\tilde{\phi}$  approximates  $f$  uniformly well on  $[0,1]^d$  except for a trifling region, we develop a technique to construct a new ReLU FNN  $\phi$  to approximate  $f$  **uniformly** well on  $[0,1]^d$  in Theorem 2.1. This technique can be applied to improve approximation errors from  $L^p$  to  $L^\infty$  for other function spaces in general, e.g., the continuous function space in [41], which is of independent interest.

In particular, if we denote the best approximation error of functions in  $C_u^s([0,1]^d)$  approximated by ReLU FNNs with width  $N$  and depth  $L$  as

$$\varepsilon_{s,d}(\tilde{N}, \tilde{L}) := \sup_{f \in C_u^s([0,1]^d)} \left( \inf_{\phi \in \mathcal{NN}(\text{width} \leq \tilde{N}; \text{depth} \leq \tilde{L})} \|\phi - f\|_{L^\infty([0,1]^d)} \right), \quad \text{for any } \tilde{N}, \tilde{L} \in \mathbb{N}^+,$$

where  $C_u^s([0,1]^d)$  denotes the unit ball of  $C^s([0,1]^d)$ . By combining the upper and lower bounds stated above, we have

$$\underbrace{C_1(s, d) \cdot \left( \tilde{N}^2 \tilde{L}^2 \ln(\tilde{N} \tilde{L}) \right)^{-s/d}}_{\text{proved in Section 2.3}} \leq \varepsilon_{s,d}(\tilde{N}, \tilde{L}) \leq \underbrace{C_2(s, d) \cdot \left( \frac{\tilde{N}^2 \tilde{L}^2}{(\ln \tilde{N} \ln \tilde{L})^2} \right)^{-s/d}}_{\text{shown in Corollary 1.2}},$$

where  $C_1(s, d)$  and  $C_2(s, d)$  are two positive constants in  $s$  and  $d$  and  $C_2(s, d)$  can be **explicitly** represented by  $s$  and  $d$ .

The expressiveness of deep neural networks has been studied extensively from many perspectives, e.g., in terms of combinatorics [34], topology [8], Vapnik-Chervonenkis (VC) dimension [7, 22, 39], fat-shattering dimension [2, 27], information theory [37], classical approximation theory [4, 5, 9, 12, 14, 15, 20, 21, 24, 29, 32, 35, 42–44, 46], etc. In the early works of approximation theory for neural networks, the universal approximation theorem [15, 23, 24] without approximation rates showed that, given any  $\varepsilon > 0$ , there exists a sufficiently large neural network approximating a target function in a certain function space within the  $\varepsilon$ -accuracy. For one-hidden-layer neural networks and functions with integral representations, Barron [5, 6] showed an asymptotic approximation rate  $\mathcal{O}(\frac{1}{\sqrt{N}})$  in the  $L^2$ -norm, leveraging an idea that is similar to Monte Carlo sampling for high-dimensional integrals. For very deep ReLU neural networks with width fixed as  $\mathcal{O}(d)$  and depth  $\mathcal{O}(L)$ , Yarotsky [44, 45] showed that the nearly optimal approximation rates for Lipschitz continuous functions and  $C^s([0,1]^d)$  functions are  $\mathcal{O}(L^{-2/d})$  and  $\mathcal{O}((L/\ln L)^{-2s/d})$ , respectively. Note that the results are asymptotic in the sense that  $L$  is required to be sufficiently large and the prefactors of these rates are unknown. To obtain a generic result that characterizes the approximation rate for arbitrary width and depth with known prefactors to guide applications, the last three authors demonstrated in [41] that the nearly optimal approximation rate for ReLU FNNs with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  to approximate Lipschitz continuous functions on  $[0,1]^d$  is  $\mathcal{O}(N^{-2/d} L^{-2/d})$ . In this paper, we extend this generic framework to  $C^s([0,1]^d)$  with a nearly optimal approximation rate  $\mathcal{O}(\|f\|_{C^s} N^{-2s/d} L^{-2s/d})$ .

Most related works are summarized in Table 1 for the comparison of our contributions in this paper and the results in the literature.

Table 1: A summary of existing approximation rates of ReLU FNNs for the Lipschitz continuous function space,  $\text{Lip}([0, 1]^d)$ , and the smooth function space,  $C^s([0, 1]^d)$ .

paper	function class	width	depth	accuracy	$L^p([0, 1]^d)$ -norm	tightness	valid for
[43]	polynomial	$\mathcal{O}(1)$	$\mathcal{O}(L)$	$\mathcal{O}(2^{-L})$	$p = \infty$		any $L \in \mathbb{N}^+$
this paper	polynomial	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}(N^{-L})$	$p = \infty$		any $N, L \in \mathbb{N}^+$
[40]	$\text{Lip}([0, 1]^d)$	$\mathcal{O}(N)$	3	$\mathcal{O}(N^{-2/d})$	$p \in [1, \infty)$	nearly tight in $N$	any $N \in \mathbb{N}^+$
[44]	$\text{Lip}([0, 1]^d)$	$2d + 10$	$\mathcal{O}(L)$	$\mathcal{O}(L^{-2/d})$	$p = \infty$	nearly tight in $L$	large $L \in \mathbb{N}^+$
[41]	$\text{Lip}([0, 1]^d)$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}(N^{-2/d}L^{-2/d})$	$p = [1, \infty]$	nearly tight in $N$ and $L$	any $N, L \in \mathbb{N}^+$
[45]	$C^s([0, 1]^d)$	$2d + 10$	$\mathcal{O}(L)$	$\mathcal{O}((L/\ln L)^{-2s/d})$	$p = \infty$	neatly tight in $L$	large $L \in \mathbb{N}^+$
this paper	$C^s([0, 1]^d)$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{O}(N^{-2s/d}L^{-2s/d})$	$p = \infty$	nearly tight in $N$ and $L$	any $N, L \in \mathbb{N}^+$
this paper	$C^s([0, 1]^d)$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}((N/\ln N)^{-2s/d}(L/\ln L)^{-2s/d})$	$p = \infty$	nearly tight in $N$ and $L$	any $N, L \in \mathbb{N}^+$

### 1.3 Discussion

We will discuss the application scope of our theory in machine learning and its comparison with existing works in more details.

#### Application scope of our theory in machine learning

In deep learning, given finitely many samples  $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^n$  of an unknown target function  $f(\mathbf{x})$  defined on a domain  $\Omega$ , a neural network  $\phi(\mathbf{x}; \boldsymbol{\theta})$  is applied to parametrize  $f$  and the best parameter set  $\boldsymbol{\theta}_S$  is identified via the following optimization problem such that  $\phi(\mathbf{x}; \boldsymbol{\theta}_S)$  can infer  $f(\mathbf{x})$ :

$$\boldsymbol{\theta}_S = \arg \min_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}), \quad \text{where } R_S(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(\phi(\mathbf{x}_i; \boldsymbol{\theta}), f(\mathbf{x}_i)) \quad (1.1)$$

with a loss function taken as  $\ell(y, y') = \frac{1}{2}|y - y'|^2$  for example. Considering the generalization to unseen data, the inference error of  $\phi(\mathbf{x}; \boldsymbol{\theta}_S)$  is usually measured by  $R_D(\boldsymbol{\theta}_S)$ , where

$$R_D(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x} \sim U(\Omega)} [\ell(\phi(\mathbf{x}; \boldsymbol{\theta}), f(\mathbf{x}))],$$

with the data distribution  $U(\Omega)$  over  $\Omega$ . In the analysis,  $U(\Omega)$  is assumed to be known, e.g, a uniform distribution for simplicity, but it is not known in real applications. In the case that  $U(\Omega)$  is a uniform distribution on  $\Omega = [0, 1]^d$  and that  $\ell(y, y') = \frac{1}{2}|y - y'|^2$ ,

$$R_D(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x} \sim U(\Omega)} [\ell(\phi(\mathbf{x}; \boldsymbol{\theta}), f(\mathbf{x}))] = \int_{[0, 1]^d} \frac{1}{2} |\phi(\mathbf{x}; \boldsymbol{\theta}) - f(\mathbf{x})|^2 d\mathbf{x}.$$

Considering all possible data following the distribution  $U(\Omega)$ , the best neural network to infer  $f(\mathbf{x})$  is actually  $\phi(\mathbf{x}; \boldsymbol{\theta}_D)$  with  $\boldsymbol{\theta}_D$  given by

$$\boldsymbol{\theta}_D = \arg \min_{\boldsymbol{\theta}} R_D(\boldsymbol{\theta}).$$

The best possible inference error is  $R_D(\boldsymbol{\theta}_D)$ . In real applications, since  $U(\Omega)$  is unknown and only finitely many samples are available, the empirical loss  $R_S(\boldsymbol{\theta})$  is minimized hoping to obtain  $\phi(\mathbf{x}; \boldsymbol{\theta}_S) \approx f(\mathbf{x})$ , instead of minimizing the population loss  $R_D(\boldsymbol{\theta})$ . When a numerical optimization method is applied to solve (1.1), it may result in a numerical solution (denoted as  $\boldsymbol{\theta}_N$ ) that is not a global minimizer. Hence, the actually

learned neural network to infer  $f(\mathbf{x})$  is  $\phi(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{N}})$  with an inference error is measured by  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$ .

Since  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$  is the expected inference error over all possible data samples, it can quantify how good the learned neural network  $\phi(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{N}})$  is. Note that

$$\begin{aligned}
R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) &= \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})]}_{\text{GE}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{OE}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\leq 0 \text{ by (1.1)}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{GE}} + \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{AE}} \\
&\leq \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{Approximation error (AE)}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{Optimization error (OE)}} + \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})] + [R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{Generalization error (GE)}}. \quad (1.2)
\end{aligned}$$

where the inequality comes from the fact that  $[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}})] \leq 0$  since  $\boldsymbol{\theta}_{\mathcal{S}}$  is a global minimizer of  $R_{\mathcal{S}}(\boldsymbol{\theta})$ . Constructive approximation provides an upper bound of  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$  in terms of the network size, e.g., in terms of the network width and depth, or in terms of the number of parameters. For example, Theorem 1.1 and its corollaries provide an upper bound  $\mathcal{O}(\|f\|_{C^s} N^{-2s/d} L^{-2s/d})$  of  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$  for  $C^s([0, 1]^d)$ . The second term of (1.2) is bounded by the optimization error of the numerical algorithm applied to solve the empirical loss minimization problem in (1.1). The study of the bounds for the third and fourth terms is referred to as the generalization error analysis of neural networks.

One of the key targets in the area of deep learning is to develop algorithms to reduce  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$ . Our analysis here provides an upper bound of the approximation error  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$  for smooth functions, which is crucial to estimate an upper bound of  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$ . Instead of deriving an approximator to attain the approximation error bound, deep learning algorithms aim at identifying a solution  $\phi(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{N}})$  reducing the generalization and optimization errors in (1.2). Solutions minimizing both generalization and optimization errors will lead to a good solution only if we also have a good upper bound estimate of  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$  as shown in (1.2). Independent of whether our analysis here leads to a good approximator, which is an interesting topic to pursue, the theory here does provide a key ingredient in the error analysis of deep learning algorithms.

We would like to emphasize that the introduction of the ReLU activation function to image classification is one of the key techniques that boost the performance of deep learning [28] with surprising generalization, which is the main reason that we focus on ReLU networks in this paper.

### Approximation rates in $\mathcal{O}(N)$ and $\mathcal{O}(L)$ versus $\mathcal{O}(W)$

It is fundamental and indispensable to characterize deep network approximation in terms of width  $\mathcal{O}(N)$ <sup>②</sup> and depth  $\mathcal{O}(L)$  simultaneously in realistic applications, while the approximation in terms of the number of nonzero parameters  $W$  is probably only of interest in theory. First of all, networks used in practice are specified via width and depth and, therefore, Theorem 1.1 can provide an error bound for such networks. However, existing results in  $W$  cannot serve for this purpose because they may be only valid for networks with other width and depth. Theories in terms of  $W$  essentially have a single variable to control the network size in three types of structures: 1) a fixed width  $N$  and a varying depth  $L$ ; 2) a fixed depth  $L$  and a varying width  $N$ ; 3) both the width and depth are controlled by the target accuracy  $\varepsilon$  (e.g.,  $N$  is a polynomial of  $\frac{1}{\varepsilon^d}$  and  $L$  is a

<sup>②</sup>For simplicity, we omit  $\mathcal{O}(\cdot)$  in the following discussion.

polynomial of  $\log(\frac{1}{\varepsilon})$ ). Therefore, given a network with an arbitrary width  $N$  and depth  $L$ , there might not be a known theory in terms of  $W$  to quantify the performance of this structure. Second, the error characterization in terms of  $N$  and  $L$  is more useful than that in terms of  $W$ , because most existing optimization and generalization analysis are based on  $N$  and  $L$  [1, 3, 10, 13, 17, 18, 25, 26], to the best of our knowledge. Approximation results in terms of  $N$  and  $L$  are more consistent with optimization and generalization analysis tools to obtain a full error analysis in (1.2).

Most existing approximation theories for deep neural networks so far focus on the approximation rate in the number of parameters  $W$  [4, 5, 9, 11, 12, 14, 15, 19–21, 24, 29–33, 35–38, 42–46]. Controlling two variables  $N$  and  $L$  in our theory is more challenging than controlling one variable  $W$  in the literature. The characterization of deep network approximation in terms of  $N$  and  $L$  can imply an approximation rate in terms of  $W$ , while it may not be true the other way around, e.g., our theorems cannot be derived from results in [45]. Let us discuss the first type of structures mentioned in the last paragraph, which includes the best-known result for a nearly optimal approximation rate,  $\mathcal{O}((W/\ln W)^{-2s/d})$ , for  $C^s$ -functions using ReLU FNNs [45], as an example to show how Theorem 1.1 in terms of  $N$  and  $L$  can be applied to show a similar result in terms of  $W$ . The main idea is to specify the value of  $N$  and  $L$  in Theorem 1.1 to show the desired corollary. For example, if we let  $N = \mathcal{O}(1)$  in Theorem 1.1, then we have the following corollary equivalent to Theorem 4.1 of [45].

**Corollary 1.3.** *Given any function  $f$  in the unit ball of  $C^s([0, 1]^d)$  with  $s \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU FNN with  $\mathcal{O}(W)$  parameters such that*

$$\|f - \phi\|_{L^\infty([0,1]^d)} \leq \mathcal{O}\left(\left(\frac{W}{\ln W}\right)^{-2s/d}\right), \quad \text{for large } W \in \mathbb{N}^+.$$

As we can see in this example, it is simple to derive Corollary 1.3 and Theorem 4.1 of [45] using Theorem 1.1 in this paper. However, Theorem 1.1 cannot be derived from any existing result that characterizes approximation rates in terms of the number of parameters. Therefore, Theorem 1.1 goes beyond existing results on the approximation of deep neural networks.

## Continuity of the weight selection

Finally, we would like to discuss the continuity of the weight selection as a map  $\Sigma : F_{s,d} \rightarrow \mathbb{R}^W$ , where  $F_{s,d}$  denotes the unit ball of the  $d$ -dimensional Sobolev space with smoothness  $s$ . For a fixed network architecture with a fixed number of parameters  $W$ , let  $g : \mathbb{R}^W \rightarrow C([0, 1]^d)$  be the map of realizing a ReLU FNN from a given set of parameters in  $\mathbb{R}^W$  to a function in  $C([0, 1]^d)$ . Suppose that the map  $\Sigma$  is continuous such that  $\|f - g(\Sigma(f))\|_{L^\infty([0,1]^d)} \leq \varepsilon$  for all  $f \in F_{s,d}$ . Then  $W \geq c\varepsilon^{-d/s}$  with some constant  $c$  depending only on  $s$ . This conclusion is given in Theorem 3 of [43], which is a corollary of Theorem 4.2 of [16] in a more general form. These theorems mean that the weight selection map  $\Sigma$  corresponding to our constructive proof in Theorem 1.1 in this paper is not continuous, since our rate is better than  $\mathcal{O}(W^{-s/d})$ . Theorem 4.2 of [16] is essentially a min-max criterion to evaluate weight selection maps maintaining continuity: the approximation error obtained by minimizing over all continuous selection  $\Sigma$  and network realization  $g$  and maximizing over all target functions is bounded below by  $\mathcal{O}(W^{-s/d})$ .



In the worst scenario, a continuous weight selection cannot enjoy an approximation rate beating  $\mathcal{O}(W^{-s/d})$ . However, Theorem 4.2 of [16] does not exclude the possibility that most functions of interest in practice may still enjoy a continuous weight selection with the approximation rate in Theorem 1.1.

**Organization:** The rest of the present paper is organized as follows. In Section 2, we prove Theorem 1.1 by combining two theorems (Theorems 2.1 and 2.2) that will be proved later. We will also discuss the optimality of Theorem 1.1 in Section 2. Next, Theorem 2.1 will be proved in Section 3 while Theorem 2.2 will be shown in Section 4. Several propositions supporting Theorem 2.2 will be presented in Section 5. Finally, Section 6 concludes this paper with a short discussion.

## 2 Approximation of smooth functions

In this section, we will prove the quantitative approximation rate in Theorem 1.1 by construction and discuss its tightness. Notations throughout the proof will be summarized in Section 2.1. The proof of Theorem 1.1 is mainly based on Theorem 2.1 and 2.2, which will be proved in Section 3 and 4, respectively. To show the tightness of Theorem 1.1, we will introduce the VC-dimension in Section 2.3.

### 2.1 Notations

Now let us summarize the main notations of the present paper as follows.

- Vectors and matrices are denoted in a bold font. Standard vectorization is adopted in matrix and vector computation. For example, a scalar plus a vector means adding the scalar to each entry of the vector. Besides, “[” and “]” are used to partition matrices (vectors) into blocks, e.g.,  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$  and  $\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} = [v_1, \dots, v_d]^T \in \mathbb{R}^d$ .
- Let  $1_S$  be the characteristic function on a set  $S$ , i.e.,  $1_S$  is equal to 1 on  $S$  and 0 outside  $S$ .
- Let  $\mathcal{B}(\mathbf{x}, r) \subseteq \mathbb{R}^d$  be the closed ball with a center  $\mathbf{x} \in \mathbb{R}^d$  and a radius  $r$ .
- Similar to “min” and “max”, let  $\text{mid}(x_1, x_2, x_3)$  be the middle value of three inputs  $x_1, x_2$ , and  $x_3$ <sup>③</sup>. For example,  $\text{mid}(2, 1, 3) = 2$  and  $\text{mid}(3, 2, 3) = 3$ .
- The set difference of two sets  $A$  and  $B$  is denoted by  $A \setminus B := \{x : x \in A, x \notin B\}$ .
- For a real number  $p \in [1, \infty)$ , the  $p$ -norm of  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d$  is defined by

$$\|\mathbf{x}\|_p := \left( |x_1|^p + |x_2|^p + \dots + |x_d|^p \right)^{1/p}.$$

---

<sup>③</sup> “mid” can be defined via  $\text{mid}(x_1, x_2, x_3) = x_1 + x_2 + x_3 - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3)$ , which can be implemented by a ReLU FNN.



- For any  $x \in \mathbb{R}$ , let  $\lfloor x \rfloor := \max\{n : n \leq x, n \in \mathbb{Z}\}$  and  $\lceil x \rceil := \min\{n : n \geq x, n \in \mathbb{Z}\}$ .
- Assume  $\mathbf{n} \in \mathbb{N}^d$ , then  $f(\mathbf{n}) = \mathcal{O}(g(\mathbf{n}))$  means that there exists positive  $C$  independent of  $\mathbf{n}$ ,  $f$ , and  $g$  such that  $f(\mathbf{n}) \leq Cg(\mathbf{n})$  when all entries of  $\mathbf{n}$  go to  $+\infty$ .

- The modulus of continuity of a continuous function  $f \in C([0, 1]^d)$  is defined as

$$\omega_f(r) := \sup \{ |f(\mathbf{x}) - f(\mathbf{y})| : \|\mathbf{x} - \mathbf{y}\|_2 \leq r, \mathbf{x}, \mathbf{y} \in [0, 1]^d \}, \quad \text{for any } r \geq 0.$$

- A  $d$ -dimensional multi-index is a  $d$ -tuple  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_d]^T \in \mathbb{N}^d$ . Several related notations are listed below.

- $\|\boldsymbol{\alpha}\|_1 = |\alpha_1| + |\alpha_2| + \dots + |\alpha_d|$ ;
- $\mathbf{x}^\boldsymbol{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d}$ , where  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$ ;
- $\boldsymbol{\alpha}! = \alpha_1! \alpha_2! \dots \alpha_d!$ ;
- $\partial^\boldsymbol{\alpha} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \dots \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}}$ .

- Given any  $K \in \mathbb{N}^+$  and  $\delta \in (0, \frac{1}{K})$ , define a trifling region  $\Omega(K, \delta, d)$  of  $[0, 1]^d$  as

$$\Omega(K, \delta, d) := \bigcup_{i=1}^d \left\{ \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_i \in \bigcup_{k=1}^{K-1} \left( \frac{k}{K} - \delta, \frac{k}{K} \right) \right\}. \quad (2.1)$$

In particular,  $\Omega(K, \delta, d) = \emptyset$  if  $K = 1$ . See Figure 1 for two examples of the trifling region.

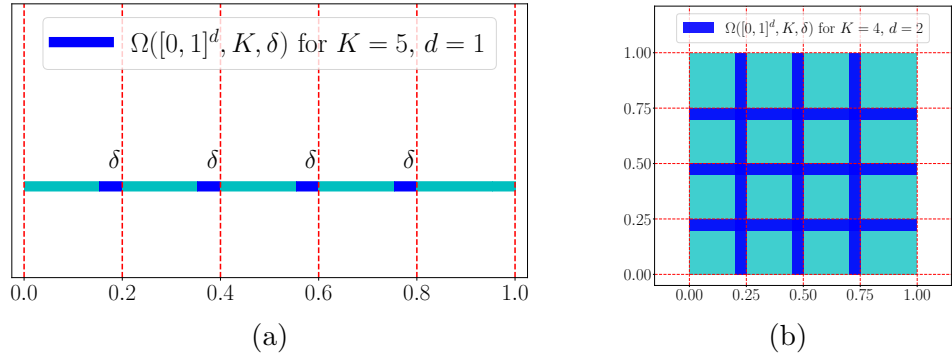


Figure 1: Two examples of the trifling region. (a)  $K = 5, d = 1$ . (b)  $K = 4, d = 2$ .

- Given  $E \subseteq \mathbb{R}^d$ , let  $C^s(E)$  denote the set containing all functions, all  $k$ -th order partial derivatives of which exist and are continuous on  $E$  for any  $k \in \mathbb{N}$  with  $0 \leq k \leq s$ . In particular,  $C^0(E)$ , also denoted by  $C(E)$ , is the set of continuous functions on  $E$ . For the case  $s = \infty$ ,  $C^\infty(E) = \bigcap_{s=0}^\infty C^s(E)$ . The  $C^s$ -norm is defined by

$$\|f\|_{C^s(E)} := \max \{ \|\partial^\boldsymbol{\alpha} f\|_{L^\infty(E)} : \boldsymbol{\alpha} \in \mathbb{N}^d \text{ with } \|\boldsymbol{\alpha}\|_1 \leq s \}.$$

Generally,  $E$  is assigned as  $[0, 1]^d$  in this paper. In particular, the closed unit ball of  $C^s([0, 1]^d)$  is denoted by

$$C_u^s([0, 1]^d) := \{ f \in C^s([0, 1]^d) : \|f\|_{C^s([0, 1]^d)} \leq 1 \}.$$

- We use “ $\mathcal{NN}$ ” as “functions implemented by ReLU FNNs” for short and use Python-type notations to specify a class of functions implemented by ReLU FNNs with several conditions. To be precise, we use  $\mathcal{NN}(c_1; c_2; \dots; c_m)$  to denote the function set containing all functions implemented by ReLU FNN architectures satisfying  $m$  conditions given by  $\{c_i\}_{1 \leq i \leq m}$ , each of which may specify the number of inputs (#input), the number of outputs (#output), the total number of nodes in all hidden layers (#neuron), the number of hidden layers (depth), the number of total parameters (#parameter), and the width in each hidden layer (widthvec), the maximum width of all hidden layers (width), etc. For example, if  $\phi \in \mathcal{NN}(\text{\#input} = 2; \text{widthvec} = [100, 100]; \text{\#output} = 1)$ , then  $\phi$  is a function satisfying the following conditions.

- $\phi$  maps from  $\mathbb{R}^2$  to  $\mathbb{R}$ .

- $\phi$  is implemented by a ReLU network with two hidden layers and the number of nodes in each hidden layer being 100.

- Let  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  denote the rectified linear unit (ReLU), i.e.  $\sigma(x) = \max\{0, x\}$ . With the abuse of notations, we define  $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$  as  $\sigma(\mathbf{x}) = \begin{bmatrix} \max\{0, x_1\} \\ \vdots \\ \max\{0, x_d\} \end{bmatrix}$  for any  $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$ .
- For a function  $\phi \in \mathcal{NN}(\text{\#input} = d; \text{widthvec} = [N_1, N_2, \dots, N_L]; \text{\#output} = 1)$ , if we set  $N_0 = d$  and  $N_{L+1} = 1$ , then the architecture of the network implementing  $\phi$  can be briefly described as follows:

$$\mathbf{x} = \tilde{\mathbf{h}}_0 \xrightarrow{\mathbf{W}_0, \mathbf{b}_0} \mathbf{h}_1 \xrightarrow{\sigma} \tilde{\mathbf{h}}_1 \dots \xrightarrow{\mathbf{W}_{L-1}, \mathbf{b}_{L-1}} \mathbf{h}_L \xrightarrow{\sigma} \tilde{\mathbf{h}}_L \xrightarrow{\mathbf{W}_L, \mathbf{b}_L} \mathbf{h}_{L+1} = \phi(\mathbf{x}),$$

where  $\mathbf{W}_i \in \mathbb{R}^{N_{i+1} \times N_i}$  and  $\mathbf{b}_i \in \mathbb{R}^{N_{i+1}}$  are the weight matrix and the bias vector in the  $i$ -th affine linear transform  $\mathcal{L}_i$  in  $\phi$ , respectively, i.e.,

$$\mathbf{h}_{i+1} = \mathbf{W}_i \cdot \tilde{\mathbf{h}}_i + \mathbf{b}_i =: \mathcal{L}_i(\tilde{\mathbf{h}}_i), \quad \text{for } i = 0, 1, \dots, L,$$

and

$$\tilde{\mathbf{h}}_i = \sigma(\mathbf{h}_i), \quad \text{for } i = 1, \dots, L.$$

In particular,  $\phi$  can be represented in a form of function compositions as follows

$$\phi = \mathcal{L}_L \circ \sigma \circ \mathcal{L}_{L-1} \circ \sigma \circ \dots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0,$$

which has been illustrated in Figure 2.

- The expression “a network (architecture) with (of) width  $N$  and depth  $L$ ” means
  - The maximum width of this network (architecture) for all **hidden** layers is no more than  $N$ .
  - The number of **hidden** layers of this network (architecture) is no more than  $L$ .
- For any  $\theta \in [0, 1)$ , suppose its binary representation is  $\theta = \sum_{\ell=1}^{\infty} \theta_{\ell} 2^{-\ell}$  with  $\theta_{\ell} \in \{0, 1\}$ , we introduce a special notation  $\text{bin}0.\theta_1\theta_2\cdots\theta_L$  to denote the  $L$ -term binary representation of  $\theta$ , i.e.,  $\text{bin}0.\theta_1\theta_2\cdots\theta_L := \sum_{\ell=1}^L \theta_{\ell} 2^{-\ell} \approx \theta$ .

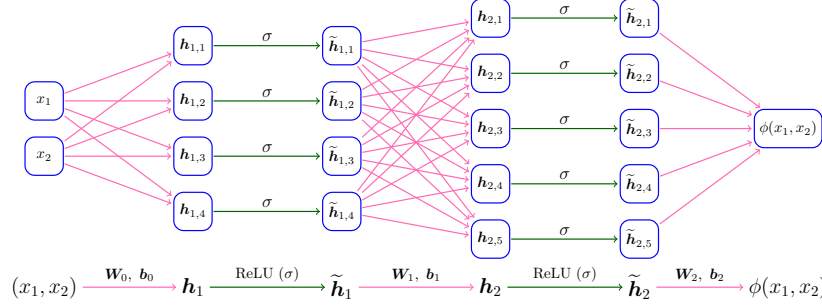


Figure 2: An example of a ReLU FNN with width 5 and depth 2.

## 2.2 Proof of Theorem 1.1

The introduction of the trifling region  $\Omega(K, \delta, d)$  is due to the fact that ReLU FNNs cannot approximate a step function uniformly well (as ReLU activation function is continuous), which is also the reason for the main difficulty of obtaining approximation rates in the  $L^\infty([0, 1]^d)$ -norm in our previous papers [40, 41]. The trifling region is a key technique to simplify the proofs of theories in [40, 41] as well as the proof of Theorem 1.1.

First, we present Theorem 2.1 to show that, as long as good uniform approximation by a ReLU FNN can be obtained outside the trifling region, the uniform approximation error can also be well controlled inside the trifling region when the network size is slightly increased. Second, as a simplified version of Theorem 1.1 ignoring the approximation error in the trifling region  $\Omega(K, \delta, d)$ , Theorem 2.2 shows the existence of a ReLU FNN approximating a target smooth function uniformly well outside the trifling region. Finally, Theorem 2.1 and 2.2 immediately lead to Theorem 1.1. Theorem 2.1 can be applied to improve the theories in [40, 41] to obtain approximation rates in the  $L^\infty([0, 1]^d)$ -norm.

**Theorem 2.1.** *Given any  $\varepsilon > 0$ ,  $N, L, K \in \mathbb{N}^+$ , and  $\delta \in (0, \frac{1}{3K}]$ , assume  $f \in C([0, 1]^d)$  and  $\tilde{\phi}$  is a function implemented by a ReLU FNN with width  $N$  and depth  $L$ . If*

$$|f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq \varepsilon, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega(K, \delta, d),$$

*then there exists a new function  $\phi$  implemented by a ReLU FNN with width  $3^d(N + 4)$  and depth  $L + 2d$  such that*

$$|f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

**Theorem 2.2.** *Assume that  $f \in C^s([0, 1]^d)$  satisfies  $\|\partial^\alpha f\|_{L^\infty([0, 1]^d)} \leq 1$  for any  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s$ . For any  $N, L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by ReLU FNN with width  $16s^{d+1}d(N + 2)\log_2(8N)$  and depth  $18s^2(L + 2)\log_2(4L)$  such that*

$$|f(\mathbf{x}) - \phi(\mathbf{x})| \leq 84(s + 1)^d 8^s N^{-2s/d} L^{-2s/d}, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

*where  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$  and  $\delta$  is an arbitrary number in  $(0, \frac{1}{3K}]$ .*

We first prove Theorem 1.1 assuming Theorem 2.1 and 2.2 are true. The proofs of Theorem 2.1 and 2.2 can be found in Section 3 and 4, respectively.

367 *Proof of Theorem 1.1.* Define  $\tilde{f} := \frac{f}{\|f\|_{C^s([0,1]^d)}} \in C_u^s([0,1]^d)$  since  $\|f\|_{C^s([0,1]^d)} = 0$  is a  
 368 trivial case. Set  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$  and choose a small  $\delta \in (0, \frac{1}{3K}]$  such that

$$369 \quad d \cdot \omega_f(\delta) \leq N^{-2s/d} L^{-2s/d}.$$

370 By Theorem 2.2, there exists a function  $\widehat{\phi}$  implemented by a ReLU FNN with width  
 371  $16s^{d+1}d(N+2)\log_2(8N)$  and depth  $18s^2(L+2)\log_2(4L)$  such that

$$372 \quad |\widehat{\phi}(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq 84(s+1)^d 8^s N^{-2s/d} L^{-2s/d}, \quad \text{for any } \mathbf{x} \in [0,1]^d \setminus \Omega([0,1]^d, K, \delta),$$

373 By Theorem 2.1, there exists a new function  $\widetilde{\phi}$  implemented by a ReLU FNN with width

$$374 \quad 3^d(16s^{d+1}d(N+2)\log_2(8N) + 4) \leq 17s^{d+1}3^d d(N+2)\log_2(8N)$$

375 and depth  $18s^2(L+2)\log_2(4L) + 2d$  such that

$$376 \quad \begin{aligned} \|\widetilde{\phi} - \tilde{f}\|_{L^\infty([0,1]^d)} &\leq 84(s+1)^d 8^s N^{-2s/d} L^{-2s/d} + d \cdot \omega_f(\delta) \\ &\leq 85(s+1)^d 8^s N^{-2s/d} L^{-2s/d}. \end{aligned}$$

377 Finally, set  $\phi = \|f\|_{C^s([0,1]^d)} \cdot \widetilde{\phi}$ , then

$$378 \quad \begin{aligned} \|\phi - f\|_{L^\infty([0,1]^d)} &= \|f\|_{C^s([0,1]^d)} \cdot \|\widetilde{\phi} - \tilde{f}\|_{L^\infty([0,1]^d)} \\ &\leq 85(s+1)^d 8^s \|f\|_{C^s([0,1]^d)} N^{-2s/d} L^{-2s/d}, \end{aligned}$$

379 and  $\phi$  can also be implemented by a ReLU FNN with width  $17s^{d+1}3^d d(N+2)\log_2(8N)$   
 380 and depth  $18s^2(L+2)\log_2(4L) + 2d$ . So we finish the proof.  $\square$

## 381 2.3 Optimality of Theorem 1.1

382 In this section, we will show that the approximation rate in Theorem 1.1 is nearly  
 383 asymptotically tight in terms of VC-dimension, denoted as  $\text{VCDim}(\mathcal{F})$  for a function  
 384 class  $\mathcal{F}$ . The key is to construct a contradiction to the VC-dimension upper bound  
 385 of ReLU FNNs in [22] if our approximation is not optimal. This idea was used in [43]  
 386 to prove its tightness for ReLU FNNs of width  $\mathcal{O}(d)$  and depth sufficiently large to  
 387 approximate smooth functions.

388 Let us first present the definitions of VC-dimension and related concepts. Let  $H$  be  
 389 a class of functions mapping from a general domain  $\mathcal{X}$  to  $\{0,1\}$ . We say  $H$  shatters the  
 390 set  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\} \subseteq \mathcal{X}$  if

$$391 \quad \left| \left\{ [h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_m)]^T \in \{0,1\}^m : h \in H \right\} \right| = 2^m,$$

392 where  $|\cdot|$  means the size of a set. This equation means, given any  $\theta_i \in \{0,1\}$  for  $i =$   
 393  $1, 2, \dots, m$ , there exists  $h \in H$  such that  $h(\mathbf{x}_i) = \theta_i$  for all  $i$ . For general a function set  $\mathcal{F}$   
 394 mapping from  $\mathcal{X}$  to  $\mathbb{R}$ , we say  $\mathcal{F}$  shatters  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\} \subseteq \mathcal{X}$  if  $\mathcal{T} \circ \mathcal{F}$  does, where

$$395 \quad \mathcal{T}(t) := \begin{cases} 1, & t \geq 0, \\ 0, & t < 0 \end{cases} \quad \text{and} \quad \mathcal{T} \circ \mathcal{F} := \{\mathcal{T} \circ f : f \in \mathcal{F}\}.$$

396 For any  $m \in \mathbb{N}^+$ , we define the growth function of  $H$  as

$$397 \quad \Pi_H(m) := \max_{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \in \mathcal{X}} \left| \left\{ [h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_m)]^T \in \{0,1\}^m : h \in H \right\} \right|.$$

**Definition 2.3** (VC-dimension). Let  $H$  be a class of functions from  $\mathcal{X}$  to  $\{0, 1\}$ . The VC-dimension of  $H$ , denoted by  $\text{VCDim}(H)$ , is the size of the largest shattered set, namely,

$$\text{VCDim}(H) := \sup \left( \{0\} \cup \{m \in \mathbb{N}^+ : \Pi_H(m) = 2^m\} \right).$$

Let  $\mathcal{F}$  be a class of functions from  $\mathcal{X}$  to  $\mathbb{R}$ . The VC-dimension of  $\mathcal{F}$ , denoted by  $\text{VCDim}(\mathcal{F})$ , is defined by  $\text{VCDim}(\mathcal{F}) := \text{VCDim}(\mathcal{T} \circ \mathcal{F})$ , where

$$\mathcal{T}(t) := \begin{cases} 1, & t \geq 0, \\ 0, & t < 0 \end{cases} \quad \text{and} \quad \mathcal{T} \circ \mathcal{F} := \{\mathcal{T} \circ f : f \in \mathcal{F}\}.$$

In particular, the expression “VC-dimension of a network (architecture)” means the VC-dimension of the function set that consists of all functions implemented by this network (architecture).

Let  $C_u^s([0, 1]^d)$  denote the unit ball of  $C^s([0, 1]^d)$  defined via

$$C_u^s([0, 1]^d) := \{f \in C^s([0, 1]^d) : \|\partial^\alpha f\|_{L^\infty([0, 1]^d)} \leq 1, \text{ for all } \alpha \in \mathbb{N}^d \text{ with } \|\alpha\|_1 \leq s\}.$$

Theorem 2.4 below shows that the best possible approximation error of functions in  $C_u^s([0, 1]^d)$  approximated by functions in  $\mathcal{F}$  is bounded by a formula characterized by  $\text{VCDim}(\mathcal{F})$ .

**Theorem 2.4.** *Given any  $s, d \in \mathbb{N}^+$ , there exists a (small) positive constant  $C_{s,d}$  determined by  $s$  and  $d$  such that: For any  $\varepsilon > 0$  and a function set  $\mathcal{F}$  with all elements defined on  $[0, 1]^d$ , if  $\text{VCDim}(\mathcal{F}) \geq 1$  and*

$$\inf_{\phi \in \mathcal{F}} \|\phi - f\|_{L^\infty([0, 1]^d)} \leq \varepsilon, \quad \text{for any } f \in C_u^s([0, 1]^d), \quad (2.2)$$

then  $\text{VCDim}(\mathcal{F}) \geq C_{s,d} \varepsilon^{-d/s}$ . <sup>④</sup>

This theorem demonstrates the connection between VC-dimension of  $\mathcal{F}$  and the approximation error using elements of  $\mathcal{F}$  to approximate functions in  $C_u^s([0, 1]^d)$ . To be precise, the best possible approximation error is controlled by  $\text{VCDim}(\mathcal{F})^{-s/d}$  up to a constant. It is shown in [22] that VC-dimension of ReLU networks with a fixed architecture with  $W$  parameters and  $L$  layers has an upper bound  $\mathcal{O}(WL \ln W)$ . It follows that VC-dimension of ReLU networks with width  $N$  and depth  $L$  is bounded by  $\mathcal{O}(N^2 L \cdot L \cdot \ln(N^2 L)) \leq \mathcal{O}(N^2 L^2 \ln(NL))$ . That is,  $\text{VCDim}(\mathcal{F}) \leq \mathcal{O}(N^2 L^2 \ln(NL))$ , where

$$\mathcal{F} = \mathcal{NN}(\# \text{input} = d; \text{width} \leq N; \text{depth} \leq L; \# \text{output} = 1).$$

Hence, the best possible approximation error of functions in  $C_u^s([0, 1]^d)$ , approximated by ReLU FNNs with width  $N$  and depth  $L$ , is

$$C(s, d) \cdot (N^2 L^2 \ln(NL))^{-s/d},$$

---

<sup>④</sup>In fact,  $C_{s,d}$  can be expressed by  $s$  and  $d$  with a **explicitly** formula as we remark in the proof of this theorem. However, the formula may be very complicated.

for some positive constant  $C(s, d)$  determined by  $s$  and  $d$ . When the width and depth become  $\mathcal{O}(N \ln N)$  and  $\mathcal{O}(L \ln L)$ , respectively, the lower bound of the approximation rate becomes

$$C(s, d) \cdot (N^2 L^2 (\ln N)^3 (\ln L)^3)^{-s/d},$$

for some positive constant  $C(s, d)$  determined by  $s$  and  $d$ . These two lower bounds mean that our approximation errors in Theorem 1.1 and Corollary 1.2 are nearly optimal. To get the lower bound

Now let us present the detailed proof of Theorem 2.4.

*Proof of Theorem 2.4.* To find a subset of  $\mathcal{F}$  shattering  $\mathcal{O}(\varepsilon^{-d/s})$  points in  $[0, 1]^d$ , we divided the proof into two steps.

- Construct  $\{f_\chi : \chi \in \mathcal{B}\} \subseteq C_u^s([0, 1]^d)$  that scatters  $\mathcal{O}(\varepsilon^{-d/s})$  points, where  $\mathcal{B}$  is a set defined later.
- Design  $\phi_\chi \in \mathcal{F}$ , for each  $\chi \in \mathcal{B}$ , based on  $f_\chi$  and Equation (2.2) such that  $\{\phi_\chi : \chi \in \mathcal{B}\} \subseteq \mathcal{F}$  also shatters  $\mathcal{O}(\varepsilon^{-d/s})$  points.

The details of these two steps can be found below.

**Step 1:** Construct  $\{f_\chi : \chi \in \mathcal{B}\} \subseteq C_u^s([0, 1]^d)$  that scatters  $\mathcal{O}(\varepsilon^{-d/s})$  points.

Let  $K = \mathcal{O}(\varepsilon^{-1/s})$  be an integer determined later and divide  $[0, 1]^d$  into  $K^d$  non-overlapping sub-cubes  $\{Q_\beta\}_\beta$  as follows:

$$Q_\beta := \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_i \in [\frac{\beta_i}{K}, \frac{\beta_i+1}{K}] \text{ for } i = 1, 2, \dots, d\},$$

for any index vector  $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T \in \{0, 1, \dots, K-1\}^d$ .

There exists  $\tilde{g} \in C^\infty(\mathbb{R}^d)$  such that  $\tilde{g}(\mathbf{0}) = 1$  and  $\tilde{g}(\mathbf{x}) = 0$  for  $\|\mathbf{x}\|_2 \geq 1/3$ .<sup>⑤</sup> Then,  $g := \tilde{g}/\tilde{C}_{s,d} \in C_u^s([0, 1]^d)$  by setting  $\tilde{C}_{s,d} := \|\tilde{g}\|_{C^s([0, 1]^d)}$ .

Define

$$\mathcal{B} := \{\chi : \chi \text{ is a map from } \{0, 1, \dots, K-1\}^d \text{ to } \{-1, 1\}\}$$

and

$$g_\beta := K^{-s} g(K(\mathbf{x} - \mathbf{x}_{Q_\beta})), \quad \text{for each } \beta \in \{0, 1, \dots, K-1\}^d,$$

where  $\mathbf{x}_{Q_\beta}$  is the center of  $Q_\beta$ .

Next, for each  $\chi \in \mathcal{B}$ , we can define  $f_\chi$  via

$$f_\chi(\mathbf{x}) := \sum_{\beta \in \{0, 1, \dots, K-1\}^d} \chi(\beta) g_\beta(\mathbf{x}).$$

Then  $f_\chi \in C_u^s([0, 1]^d)$  for each  $\chi \in \mathcal{B}$ , since it satisfies the following two conditions.

- By the definition of  $g_\beta$  and  $\chi$ , we have

$$\{\mathbf{x} : \chi(\beta) g_\beta(\mathbf{x}) \neq 0\} \subseteq \mathcal{B}(\mathbf{x}_{Q_\beta}, \frac{1}{3K}) \subseteq \frac{2}{3} Q_\beta, \quad \text{for each } \beta \in \{0, 1, \dots, K-1\}^d.$$

<sup>⑤</sup>In fact, such a  $\tilde{g}$  is called “bump function”. An example can be attained by setting  $\tilde{g}(\mathbf{x}) = C \exp(\frac{1}{\|3\mathbf{x}\|_2^2 - 1})$  if  $\|\mathbf{x}\|_2 < 1/3$  and  $\tilde{g}(\mathbf{x}) = 0$  if  $\|\mathbf{x}\|_2 \geq 1/3$ , where  $C$  is a proper constant such that  $\tilde{g}(\mathbf{0}) = 1$ .

• For any  $\mathbf{x} \in Q_\beta$ ,  $\beta \in \{0, 1, \dots, K-1\}^d$ , and  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s$ ,

$$\partial^\alpha f_\chi(\mathbf{x}) = \chi(\beta) \partial^\alpha g_\beta(\mathbf{x}) = K^{-s} \chi(\beta) K^{\|\alpha\|_1} \partial^\alpha g(K(\mathbf{x} - \mathbf{x}_\beta)),$$

which implies  $|\partial^\alpha f_\chi(\mathbf{x})| = |K^{-(s-\|\alpha\|_1)} \partial^\alpha g(K(\mathbf{x} - \mathbf{x}_\beta))| \leq 1$ .

It is easy to check that  $\{f_\chi : \chi \in \mathcal{B}\} \subseteq C_u^s([0, 1]^d)$  can shatter  $K^d = \mathcal{O}(\varepsilon^{-d/\alpha})$  points in  $[0, 1]^d$ .

**Step 2:** Construct  $\{\phi_\chi : \chi \in \mathcal{B}\}$  that also scatters  $\mathcal{O}(\varepsilon^{-d/s})$  points.

By Equation (2.2), for each  $\chi \in \mathcal{B}$ , there exists  $\phi_\chi \in \mathcal{F}$  such that

$$\|\phi_\chi - f_\chi\|_{L^\infty([0, 1]^d)} \leq \varepsilon + \varepsilon/2.$$

Let  $\mu(\cdot)$  denote the Lebesgue measure of a set. Then, for each  $\chi \in \mathcal{B}$ , there exists  $\mathcal{H}_\chi \subseteq [0, 1]^d$  with  $\mu(\mathcal{H}_\chi) = 0$  such that

$$|\phi_\chi(\mathbf{x}) - f_\chi(\mathbf{x})| \leq \frac{3}{2}\varepsilon, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}_\chi.$$

Set  $\mathcal{H} = \cup_{\chi \in \mathcal{B}} \mathcal{H}_\chi$ , then we have  $\mu(\mathcal{H}) = 0$  and

$$|\phi_\chi(\mathbf{x}) - f_\chi(\mathbf{x})| \leq \frac{3}{2}\varepsilon, \quad \text{for any } \chi \in \mathcal{B} \text{ and } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}. \quad (2.3)$$

Clearly, there exists  $r \in (0, 1)$  such that

$$g_\beta(\mathbf{x}) \geq \frac{1}{2}g_\beta(\mathbf{x}_{Q_\beta}), \quad \text{for any } \mathbf{x} \in rQ_\beta,$$

where  $\mathbf{x}_{Q_\beta}$  is the center of  $Q_\beta$ .

Note that  $(rQ_\beta) \setminus \mathcal{H}$  is not empty, since  $\mu((rQ_\beta) \setminus \mathcal{H}) > 0$  for each  $\beta$ . Then, for each  $\chi \in \mathcal{B}$  and  $\beta \in \{0, 1, \dots, K-1\}^d$ , there exists  $\mathbf{x}_\beta \in (rQ_\beta) \setminus \mathcal{H}$  such that

$$|f_\chi(\mathbf{x}_\beta)| = |g_\beta(\mathbf{x}_\beta)| \geq \frac{1}{2}|g_\beta(\mathbf{x}_{Q_\beta})| = \frac{1}{2}K^{-s}g(\mathbf{0}) = \frac{1}{2}K^{-s}/\tilde{C}_{s,d} \geq 2\varepsilon, \quad (2.4)$$

where the last inequality is attained by setting  $K = \lfloor (4\varepsilon\tilde{C}_{s,d})^{-1/s} \rfloor$ . Note that it is necessary to verify  $K \neq 0$ , we do it later in the proof.

By Equation (2.3) and (2.4), we have, for each  $\beta \in \{0, 1, \dots, K-1\}^d$  and each  $\chi \in \mathcal{B}$ ,

$$|f_\chi(\mathbf{x}_\beta)| \geq 2\varepsilon > \frac{3}{2}\varepsilon \geq |\phi_\chi(\mathbf{x}_\beta) - \phi_\chi(\mathbf{x}_\beta)|.$$

So,  $f_\chi(\mathbf{x}_\beta)$  and  $\phi_\chi(\mathbf{x}_\beta)$  have the same sign for each  $\chi \in \mathcal{B}$  and  $\beta \in \{0, 1, \dots, K-1\}^d$ . Then  $\{\phi_\chi : \chi \in \mathcal{B}\}$  shatters  $\{\mathbf{x}_\beta : \beta \in \{0, 1, \dots, K-1\}^d\}$  since  $\{f_\chi : \chi \in \mathcal{B}\}$  shatters  $\{\mathbf{x}_\beta : \beta \in \{0, 1, \dots, K-1\}^d\}$ . Hence,

$$\text{VCDim}(\mathcal{F}) \geq \text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \geq K^d = \lfloor (4\varepsilon\tilde{C}_{s,d})^{-1/s} \rfloor^d \geq 2^{-d}(4\varepsilon\tilde{C}_{s,d})^{-d/s},$$

where the last inequality comes from the fact  $\lfloor x \rfloor \geq x/2$  for any  $x \in [1, \infty)$ .

Finally, by setting

$$C_{s,d} = 2^{-d}(4\tilde{C}_{s,d})^{-d/s} = 2^{-d}(4\|\tilde{g}\|_{C^s([0, 1]^d)})^{-d/s},$$



we have

$$\text{VCDim}(\mathcal{F}) \geq 2^{-d} (4\varepsilon \tilde{C}_{s,d})^{-d/s} = C_{s,d} \varepsilon^{-d/s}$$

and

$$K = \lfloor (4\varepsilon \tilde{C}_{s,d})^{-1/s} \rfloor = \lfloor \varepsilon^{-1/s} (2C_{s,d})^{1/d} \rfloor \geq 1,$$

where the last inequality comes from the assumption  $\varepsilon \leq (2^d C_{s,d})^{s/d}$ . Such an assumption is reasonable since  $\varepsilon > (2^d C_{s,d})^{s/d}$  is a trivial case, which implies

$$\text{VCDim}(\mathcal{F}) \geq 1 \geq 2^{-d} = C_{s,d} \left( (2^d C_{s,d})^{s/d} \right)^{-d/s} > C_{s,d} \varepsilon^{-d/s}.$$

So we finish the proof.  $\square$

### 3 Proof of Theorem 2.1

Intuitively speaking, Theorem 2.1 shows that: if a ReLU FNN  $g$  approximates  $f$  well except for a trifling region, then we can extend  $g$  to approximate  $f$  well on the whole domain. For example, if  $g$  approximates a one-dimensional continuous function  $f$  well except for a region in  $\mathbb{R}$  with a sufficiently small measure  $\delta$ , then  $\text{mid}(g(x+\delta), g(x), g(x-\delta))$  can approximate  $f$  well on the whole domain, where  $\text{mid}(\cdot, \cdot, \cdot)$  is a function returning the middle value of three inputs and can be implemented via a ReLU FNN as shown in Lemma 3.1. This key idea is called the horizontal shift (translation) of  $g$  in this paper.

**Lemma 3.1.** *The middle value function  $\text{mid}(x_1, x_2, x_3)$  can be implemented by a ReLU FNN  $\phi$  with width 14 and depth 2.*

*Proof.* Recall the fact

$$x = \sigma(x) - \sigma(-x) \quad \text{and} \quad |x| = \sigma(x) + \sigma(-x), \quad \text{for any } x \in \mathbb{R}. \quad (3.1)$$

Therefore,

$$\begin{aligned} \max(x, y) &= \frac{x + y + |x - y|}{2} \\ &= \frac{1}{2} \sigma(x + y) - \frac{1}{2} \sigma(-x - y) + \frac{1}{2} \sigma(x - y) + \frac{1}{2} \sigma(-x + y), \end{aligned} \quad (3.2)$$

for any  $x, y \in \mathbb{R}$ . Thus,  $\max(x_1, x_2, x_3)$  can be implemented by the network shown in Figure 3.

Clearly,

$$\max(x_1, x_2, x_3) \in \mathcal{NN}(\# \text{input} = 3; \text{widthvec} = [6, 4]).$$

Similarly, we have

$$\min(x_1, x_2, x_3) \in \mathcal{NN}(\# \text{input} = 3; \text{widthvec} = [6, 4]).$$

It is easy to check that

$$\begin{aligned} \text{mid}(x_1, x_2, x_3) &= x_1 + x_2 + x_3 - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3) \\ &= \sigma(x_1 + x_2 + x_3) - \sigma(-x_1 - x_2 - x_3) - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3). \end{aligned}$$

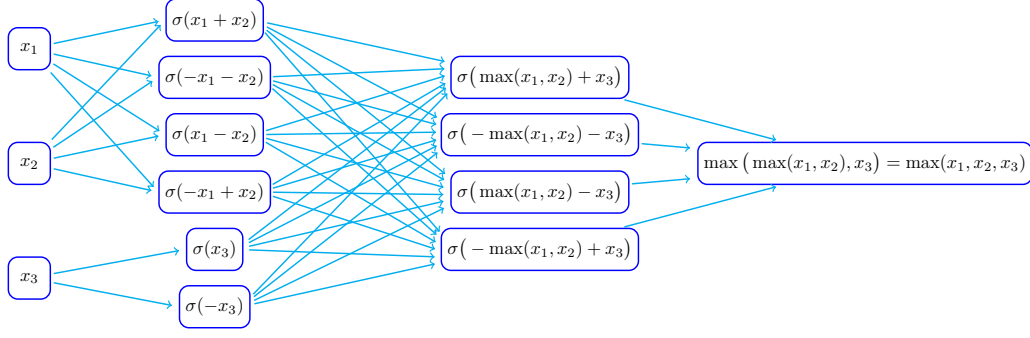


Figure 3: An illustration of the network architecture implementing  $\max(x_1, x_2, x_3)$  based on Equation (3.1) and (3.2).

Hence,

$$\text{mid}(x_1, x_2, x_3) \in \mathcal{NN}(\#input = 3; \text{widthvec} = [14, 10]),$$

that is,  $\text{mid}(x_1, x_2, x_3)$  can be implemented by a ReLU FNN  $\phi$  with width 14 and depth 2. So we finish the proof.  $\square$

The next lemma shows a simple but useful property of the  $\text{mid}(x_1, x_2, x_3)$  function that helps to exclude poor approximation in the trifling region.

**Lemma 3.2.** *For any  $\varepsilon > 0$ , if at least two of  $\{x_1, x_2, x_3\}$  are in  $\mathcal{B}(y, \varepsilon)$ , then  $\text{mid}(x_1, x_2, x_3) \in \mathcal{B}(y, \varepsilon)$ .*

*Proof.* Without loss of generality, we may assume  $x_1, x_2 \in \mathcal{B}(y, \varepsilon)$  and  $x_1 \leq x_2$ . Then the proof can be divided into three cases.

1. If  $x_3 < x_1$ , then  $\text{mid}(x_1, x_2, x_3) = x_1 \in \mathcal{B}(y, \varepsilon)$ .
2. If  $x_1 \leq x_3 \leq x_2$ , then  $\text{mid}(x_1, x_2, x_3) = x_3 \in \mathcal{B}(y, \varepsilon)$  since  $y - \varepsilon \leq x_1 \leq x_3 \leq x_2 \leq y + \varepsilon$ .
3. If  $x_2 < x_3$ , then  $\text{mid}(x_1, x_2, x_3) = x_2 \in \mathcal{B}(y, \varepsilon)$ .

So we finish the proof.  $\square$

Next, given a function  $g$  approximating  $f$  well on  $[0, 1]$  except for a trifling region, Lemma 3.3 below shows how to use the  $\text{mid}(x_1, x_2, x_3)$  function to construct a new function  $\phi$  uniformly approximating  $f$  well on  $[0, 1]$ , leveraging the useful property of  $\text{mid}(x_1, x_2, x_3)$  in Lemma 3.2.

**Lemma 3.3.** *Given any  $\varepsilon > 0$ ,  $K \in \mathbb{N}^+$ , and  $\delta \in (0, \frac{1}{3K}]$ , assume  $f \in C([0, 1])$  and  $g : \mathbb{R} \rightarrow \mathbb{R}$  is a general function with*

$$|g(x) - f(x)| \leq \varepsilon, \text{ i.e., } g(x) \in \mathcal{B}(f(x), \varepsilon), \quad \text{for any } x \in [0, 1] \setminus \Omega([0, 1], K, \delta). \quad (3.3)$$

Then

$$|\phi(x) - f(x)| \leq \varepsilon + \omega_f(\delta), \quad \text{for any } x \in [0, 1],$$

where

$$\phi(x) := \text{mid}(g(x - \delta), g(x), g(x + \delta)), \quad \text{for any } x \in \mathbb{R}.$$

547 *Proof.* Divide  $[0, 1]$  into  $K$  small intervals denoted by  $Q_k = [\frac{k}{K}, \frac{k+1}{K}]$  for  $k = 0, 1, \dots, K-1$ .  
 548 For each  $k$ , we further divide  $Q_k$  into four small closed intervals as shown in Figure 4.  
 549 To be exact,

$$550 \quad Q_k = Q_{k,1} \cup Q_{k,2} \cup Q_{k,3} \cup Q_{k,4},$$

551 where  $Q_{k,1} = [\frac{k}{K}, \frac{k}{K} + \delta]$ ,  $Q_{k,2} = [\frac{k}{K} + \delta, \frac{k+1}{K} - 2\delta]$ ,  $Q_{k,3} = [\frac{k+1}{K} - 2\delta, \frac{k+1}{K} - \delta]$ , and  $Q_{k,4} =$   
 552  $[\frac{k+1}{K} - \delta, \frac{k+1}{K}]$ .

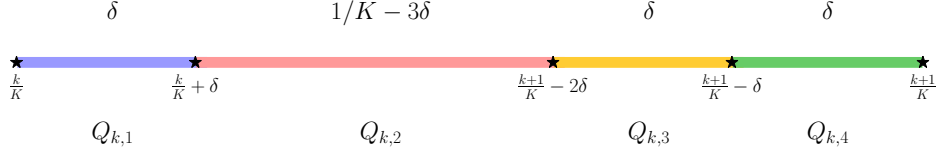


Figure 4: An illustration of  $Q_{k,i}$  for  $i = 1, 2, 3, 4$ .

553 Clearly,  $Q_{K-1,4} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta)$  and  $Q_{k,i} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta)$  for  $k = 0, 1, \dots, k-$   
 554  $1$  and  $i = 1, 2, 3$ .

555 To estimate the difference between  $\phi(x)$  and  $f(x)$ , we consider the following four  
 556 cases of  $x$  in  $[0, 1]$  for  $k \in \{0, 1, \dots, K-1\}$ .

557 **Case 1:**  $x \in Q_{k,1}$ .

558 If  $x \in Q_{k,1}$ , then  $x \in [0, 1] \setminus \Omega([0, 1], K, \delta)$  and

$$559 \quad x + \delta \in Q_{k,2} \cup Q_{k,3} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta).$$

560 It follows from Equation (3.3) that

$$561 \quad g(x) \in \mathcal{B}(f(x), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta))$$

562 and

$$563 \quad g(x + \delta) \in \mathcal{B}(f(x + \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

564 By Lemma 3.2, we get

$$565 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

566 **Case 2:**  $x \in Q_{k,2}$ .

567 If  $x \in Q_{k,2}$ , then  $x - \delta, x, x + \delta \in [0, 1] \setminus \Omega([0, 1], K, \delta)$ . It follows from Equation (3.3)  
 568 that

$$569 \quad g(x - \delta) \in \mathcal{B}(f(x - \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)),$$

570

$$571 \quad g(x) \in \mathcal{B}(f(x), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)),$$

572 and

$$573 \quad g(x + \delta) \in \mathcal{B}(f(x + \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta))$$

574 Then, by Lemma 3.2, we have

$$575 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

576 **Case 3:**  $x \in Q_{k,3}$ .

577 If  $x \in Q_{k,3}$ , then  $x \in [0, 1] \setminus \Omega([0, 1], K, \delta)$  and

$$578 \quad x - \delta \in Q_{k,1} \cup Q_{k,2} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta).$$

579 It follows from Equation (3.3) that

$$580 \quad g(x) \in \mathcal{B}(f(x), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta))$$

581 and

$$582 \quad g(x - \delta) \in \mathcal{B}(f(x - \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

583 By Lemma 3.2, we get

$$584 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

585 **Case 4:**  $x \in Q_{k,4}$ .

586 If  $x \in Q_{k,4}$ , we can divide this case into two sub-cases.

- 587 • If  $k \in \{0, 1, \dots, K - 2\}$ , then  $x - \delta \in Q_{k,3} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta)$  and  $x + \delta \in Q_{k+1,1} \subseteq$   
 588  $[0, 1] \setminus \Omega([0, 1], K, \delta)$ . It follows from Equation (3.3) that

$$589 \quad g(x - \delta) \in \mathcal{B}(f(x - \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta))$$

590 and

$$591 \quad g(x + \delta) \in \mathcal{B}(f(x + \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

592 By Lemma 3.2, we get

$$593 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

- 594 • If  $k = K - 1$ , then  $x \in Q_{k,4} = Q_{K-1,4} \subseteq [0, 1] \setminus \Omega([0, 1], K, \delta)$  and  $x - \delta \in Q_{k,3} \subseteq$   
 595  $[0, 1] \setminus \Omega([0, 1], K, \delta)$ . It follows from Equation (3.3) that

$$596 \quad g(x) \in \mathcal{B}(f(x), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta))$$

597 and

$$598 \quad g(x - \delta) \in \mathcal{B}(f(x - \delta), \varepsilon) \subseteq \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

599 By Lemma 3.2, we get

$$600 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)).$$

601 Since  $[0, 1] = \cup_{k=0}^{K-1} \left( \cup_{i=1}^4 Q_{k,i} \right)$ , we have

$$602 \quad \text{mid}(g(x - \delta), g(x), g(x + \delta)) \in \mathcal{B}(f(x), \varepsilon + \omega_f(\delta)), \quad \text{for any } x \in [0, 1].$$

603 Recall that  $\phi(x) = \text{mid}(g(x - \delta), g(x), g(x + \delta))$ . Then we have

$$604 \quad |\phi(x) - f(x)| \leq \varepsilon + \omega_f(\delta), \quad \text{for any } x \in [0, 1].$$

605 So we finish the proof. □

606 The next lemma below is an analog of Lemma 3.3.

607 **Lemma 3.4.** *Given any  $\varepsilon > 0$ ,  $K \in \mathbb{N}^+$ , and  $\delta \in (0, \frac{1}{3K}]$ , assume  $f \in C([0, 1]^d)$  and*  
 608  *$g : \mathbb{R}^d \rightarrow \mathbb{R}$  is a general function with*

$$609 \quad |g(\mathbf{x}) - f(\mathbf{x})| \leq \varepsilon, \text{ i.e., } g(\mathbf{x}) \in \mathcal{B}(f(\mathbf{x}), \varepsilon), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta).$$

610 Then

$$611 \quad |\phi(\mathbf{x}) - f(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d,$$

612 where  $\phi := \phi_d$  is defined by induction through

$$613 \quad \phi_{i+1}(\mathbf{x}) := \text{mid}(\phi_i(\mathbf{x} - \delta \mathbf{e}_{i+1}), \phi_i(\mathbf{x}), \phi_i(\mathbf{x} + \delta \mathbf{e}_{i+1})), \quad \text{for } i = 0, 1, \dots, d-1, \quad (3.4)$$

614 where  $\phi_0$  is equal to  $g$  and  $\{\mathbf{e}_i\}_{i=1}^d$  is the standard basis in  $\mathbb{R}^d$ .

615 *Proof.* For  $\ell = 0, 1, \dots, d$ , we define

$$616 \quad E_\ell := \left\{ \mathbf{x} = [x_1, x_2, \dots, x_d]^T : x_i \in \begin{cases} [0, 1], & \text{if } i \leq \ell, \\ [0, 1] \setminus \Omega([0, 1], K, \delta), & \text{if } i > \ell \end{cases} \right\}.$$

617 Clearly,  $E_0 = [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$  and  $E_d = [0, 1]^d$ . See Figure 5 for the illustrations of  $E_\ell$  for  $\ell = 0, 1, \dots, d$  when  $K = 4$  and  $d = 2$ .

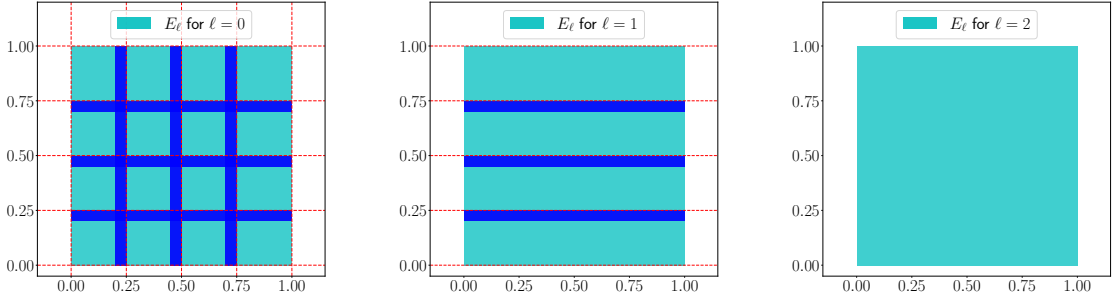


Figure 5: Illustrations of  $E_\ell$  for  $\ell = 0, 1, 2$  when  $K = 4$  and  $d = 2$ .

618

619 We would like to construct a sequence of functions  $\phi_0, \phi_1, \dots, \phi_d$  by induction, based  
 620 on the iteration Equation (3.4), such that, for each  $\ell \in \{0, 1, \dots, d\}$ ,

$$621 \quad \phi_\ell(\mathbf{x}) \in \mathcal{B}(f(\mathbf{x}), \varepsilon + \ell \cdot \omega_f(\delta)), \quad \text{for any } \mathbf{x} \in E_\ell. \quad (3.5)$$

622 Let us first consider the case  $\ell = 0$ . Note that  $\phi_0$  is a extension of  $g \in C([0, 1]^d)$ ,  
 623  $E_0 = [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ , and  $|g(\mathbf{x}) - f(\mathbf{x})| \leq \varepsilon$  for any  $\mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ .  
 624 Then we have

$$625 \quad \phi_0(\mathbf{x}) = g(\mathbf{x}) \in \mathcal{B}(f(\mathbf{x}), \varepsilon), \quad \text{for any } \mathbf{x} \in E_0.$$

626 That is, Equation (3.5) is true for  $\ell = 0$ .

627 Now assume Equation (3.5) is true for  $\ell = i$ . We will prove that it also holds for  
 628  $\ell = i + 1$ . By the hypothesis of induction, we have

$$629 \quad \phi_i(x_1, \dots, x_i, t, x_{i+2}, \dots, x_d) \in \mathcal{B}(f(x_1, \dots, x_i, t, x_{i+2}, \dots, x_d), \varepsilon + i \cdot \omega_f(\delta)), \quad (3.6)$$

630 for any  $x_1, \dots, x_i \in [0, 1]$  and  $t, x_{i+2}, \dots, x_d \in [0, 1] \setminus \Omega([0, 1], K, \delta)$ .

631 Fix  $x_1, \dots, x_i \in [0, 1]$  and  $x_{i+2}, \dots, x_d \in [0, 1] \setminus \Omega([0, 1], K, \delta)$ , and denote

$$632 \quad \mathbf{x}^{[i]} := [x_1, \dots, x_i, x_{i+2}, \dots, x_d]^T.$$

633 Then define

$$634 \quad \psi_{\mathbf{x}^{[i]}}(t) := \phi_i(x_1, \dots, x_i, t, x_{i+2}, \dots, x_d), \quad \text{for any } t \in \mathbb{R},$$

635 and

$$636 \quad f_{\mathbf{x}^{[i]}}(t) := f(x_1, \dots, x_i, t, x_{i+2}, \dots, x_d), \quad \text{for any } t \in \mathbb{R}.$$

637 It follows from Equation (3.6) that

$$638 \quad \psi_{\mathbf{x}^{[i]}}(t) \in \mathcal{B}(f_{\mathbf{x}^{[i]}}(t), \varepsilon + i \cdot \omega_f(\delta)), \quad \text{for any } t \in [0, 1] \setminus \Omega([0, 1], K, \delta).$$

639 Then by Lemma 3.3 (set  $g = \psi_{\mathbf{x}^{[i]}}$  and  $f = f_{\mathbf{x}^{[i]}}$  therein), we get, for any  $t \in [0, 1]$ ,

$$640 \quad \begin{aligned} \text{mid}(\psi_{\mathbf{x}^{[i]}}(t - \delta), \psi_{\mathbf{x}^{[i]}}(t), \psi_{\mathbf{x}^{[i]}}(t + \delta)) &\in \mathcal{B}(f_{\mathbf{x}^{[i]}}(t), \varepsilon + i \cdot \omega_f(\delta) + \omega_{f_{\mathbf{x}^{[i]}}}(\delta)) \\ &\subseteq \mathcal{B}(f_{\mathbf{x}^{[i]}}(t), \varepsilon + (i + 1)\omega_f(\delta)). \end{aligned}$$

641 That is, for any  $x_{i+1} = t \in [0, 1]$ ,

$$642 \quad \begin{aligned} &\text{mid}(\phi_i(x_1, \dots, x_i, x_{i+1} - \delta, x_{i+2}, \dots, x_d), \phi_i(x_1, \dots, x_d), \phi_i(x_1, \dots, x_i, x_{i+1} + \delta, x_{i+2}, \dots, x_d)) \\ &\in \mathcal{B}(f(x_1, \dots, x_d), \varepsilon + (i + 1)\omega_f(\delta)). \end{aligned}$$

643 Since  $x_1, \dots, x_i \in [0, 1]$  and  $x_{i+2}, \dots, x_d \in [0, 1] \setminus \Omega([0, 1], K, \delta)$  are arbitrary, then for any  
644  $\mathbf{x} \in E_{i+1}$ ,

$$645 \quad \text{mid}(\phi_i(\mathbf{x} - \delta \mathbf{e}_{i+1}), \phi_i(\mathbf{x}), \phi_i(\mathbf{x} + \delta \mathbf{e}_{i+1})) \in \mathcal{B}(f(\mathbf{x}), \varepsilon + (i + 1)\omega_f(\delta)),$$

646 which implies

$$647 \quad \phi_{i+1}(\mathbf{x}) \in \mathcal{B}(f(\mathbf{x}), \varepsilon + (i + 1)\omega_f(\delta)), \quad \text{for any } \mathbf{x} \in E_{i+1}.$$

648 So we show that Equation (3.5) is true for  $\ell = i + 1$ , which means we finish the process  
649 of mathematical induction.

650 By the principle of induction, we have

$$651 \quad \phi(\mathbf{x}) := \phi_d(\mathbf{x}) \in \mathcal{B}(f(\mathbf{x}), \varepsilon + d \cdot \omega_f(\delta)), \quad \text{for any } \mathbf{x} \in E_d = [0, 1]^d.$$

652 Therefore,

$$653 \quad |\phi(\mathbf{x}) - f(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d,$$

654 which means we finish the proof. □

655 Now we are ready to prove Theorem 2.1.

656 *Proof of Theorem 2.1.* Set  $\phi_0 = \tilde{\phi}$  and define  $\phi_i$  for  $i \in \{1, \dots, d-1\}$  by induction as follows:

657 
$$\phi_{i+1}(\mathbf{x}) := \text{mid}(\phi_i(\mathbf{x} - \delta \mathbf{e}_{i+1}), \phi_i(\mathbf{x}), \phi_i(\mathbf{x} + \delta \mathbf{e}_{i+1})), \quad \text{for } i = 0, 1, \dots, d-1,$$

658 where  $\{\mathbf{e}_i\}_{i=1}^d$  is the standard basis in  $\mathbb{R}^d$ . Then by Lemma 3.4 with  $\phi = \phi_d$ , we have

659 
$$|\phi(\mathbf{x}) - f(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

660 It remains to determine the network architecture implementing  $\phi = \phi_d$ . Clearly,  $\phi_0 = \tilde{\phi} \in$   
661  $\mathcal{NN}(\text{width} \leq N; \text{depth} \leq L)$  implies

662 
$$\phi_0(\cdot - \delta \mathbf{e}_1), \phi_0(\cdot), \phi_0(\cdot + \delta \mathbf{e}_1) \in \mathcal{NN}(\text{width} \leq N; \text{depth} \leq L).$$

663 By defining a vector-valued function  $\Phi_0 : \mathbb{R}^d \rightarrow \mathbb{R}^3$  as

664 
$$\Phi_0(\mathbf{x}) := (\phi_0(\mathbf{x} - \delta \mathbf{e}_1), \phi_0(\mathbf{x}), \phi_0(\mathbf{x} + \delta \mathbf{e}_1)), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d,$$

665 we have  $\Phi_0 \in \mathcal{NN}(\#input = d; \text{width} \leq 3N; \text{depth} \leq L; \#output = 3)$ . Recall that  
666  $\text{mid}(\cdot, \cdot, \cdot) \in \mathcal{NN}(\text{width} \leq 14; \text{depth} \leq 2)$  by Lemma 3.1. Therefore,  $\phi_1 = \min(\cdot, \cdot, \cdot) \circ \Phi_0$   
667 can be implemented by a ReLU FNN with width  $\max\{3N, 14\} \leq 3(N+4)$  and depth  
668  $L+2$ . Similarly,  $\phi = \phi_d$  can be implemented by a ReLU FNN with width  $3^d(N+4)$  and  
669 depth  $L+2d$ . So we finish the proof.  $\square$

## 670 4 Proof of Theorem 2.2

671 In this section, we prove Theorem 2.2, a weaker version of the main theorem of this  
672 paper (Theorem 1.1) targeting a ReLU FNN constructed to approximate a smooth func-  
673 tion outside the trifling region. The main idea is to construct ReLU FNNs through Taylor  
674 expansions of smooth functions. We first discuss the sketch of the proof in Section 4.1  
675 and give the detailed proof in Section 4.2.

### 676 4.1 Sketch of the proof of Theorem 2.2

677 Set  $K = \mathcal{O}(N^{2/d} L^{2/d})$  and let  $\Omega([0, 1]^d, K, \delta)$  partition  $[0, 1]^d$  into  $K^d$  cubes  $Q_{\beta}$   
678 for  $\beta \in \{0, 1, \dots, K-1\}^d$ . As we shall see later, the introduction of the trifling region  
679  $\Omega([0, 1]^d, K, \delta)$  can reduce the difficulty to construct ReLU FNNs to achieve the op-  
680 timal approximation rate simultaneously in width and depth, since it is only required  
681 to uniformly control the approximation error outside the trifling region and there is  
682 no requirement for the ReLU FNN inside the trifling region. In particular, for each  
683  $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T \in \{0, 1, \dots, K-1\}^d$ , we define  $\mathbf{x}_{\beta} := \beta/K$  and

684 
$$Q_{\beta} = \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T : x_i \in [\frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \leq K-2\}}]\} \text{ for } i = 1, 2, \dots, d\}.$$

685 Clearly,  $[0, 1]^d = \Omega([0, 1]^d, K, \delta) \cup (\cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_{\beta})$  and  $\mathbf{x}_{\beta}$  is the vertex of  $Q_{\beta}$  with  
686 minimum  $\|\cdot\|_1$  norm. See Figure 6 for the illustrations of  $Q_{\beta}$  and  $\mathbf{x}_{\beta}$ .

687 For any  $\beta \in \{0, 1, \dots, K-1\}^d$  and  $\mathbf{x} \in Q_{\beta}$ , there exists  $\xi_{\mathbf{x}} \in (0, 1)$  such that

688 
$$f(\mathbf{x}) = \sum_{\|\alpha\|_1 \leq s-1} \frac{\partial^{\alpha} f(\mathbf{x}_{\beta})}{\alpha!} \mathbf{h}^{\alpha} + \sum_{\|\alpha\|_1 = s} \frac{\partial^{\alpha} f(\mathbf{x}_{\beta} + \xi_{\mathbf{x}} \mathbf{h})}{\alpha!} \mathbf{h}^{\alpha} =: \mathcal{T}_1 + \mathcal{T}_2, \quad \textcircled{6} \quad (4.1)$$

---

$\textcircled{6} \sum_{\|\alpha\|_1 = s}$  is short for  $\sum_{\|\alpha\|_1 = s, \alpha \in \mathbb{N}^d}$ . The same notation is used throughout this paper.



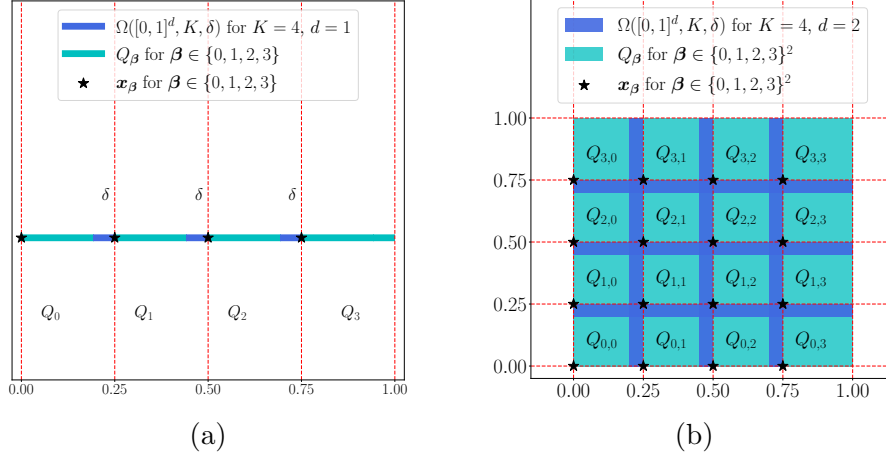


Figure 6: Illustrations of  $\Omega([0, 1]^d, K, \delta)$ ,  $Q_\beta$ , and  $\mathbf{x}_\beta$  for  $\beta \in \{0, 1, \dots, K-1\}^d$ . (a)  $K = 4$  and  $d = 1$ . (b)  $K = 4$  and  $d = 2$ .

where  $\mathbf{h}(\mathbf{x}) = \mathbf{x} - \mathbf{x}_\beta = \mathbf{x} - \beta/K$ . It is clear that the magnitude of  $\mathcal{T}_2$  is bounded by  $\mathcal{O}(K^{-s}) = \mathcal{O}(N^{-2s/d} L^{-2s/d})$ . So we only need to construct a function in  $\mathcal{NN}$  (width  $\leq \mathcal{O}(N \ln N)$ ; depth  $\leq \mathcal{O}(L \ln L)$ ) to approximate

$$\mathcal{T}_1 = \sum_{\|\alpha\|_1 \leq s-1} \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha$$

within an error  $\mathcal{O}(N^{-2s/d} L^{-2s/d})$ . To approximate  $\mathcal{T}_1$  well by ReLU FNNs, we need three key steps as follows.

- Construct a ReLU FNN to implement a vector-valued function  $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}^d$  projecting the whole cube  $Q_\beta$  to the point  $\mathbf{x}_\beta = \beta/K$ , i.e.,  $\Psi(\mathbf{x}) = \mathbf{x}_\beta$  for any  $\mathbf{x} \in Q_\beta$  and each  $\beta \in \{0, 1, \dots, K-1\}^d$ .
- Construct a ReLU FNN to implement a function  $P_\alpha : \mathbb{R}^d \rightarrow \mathbb{R}$  approximating the polynomial  $\mathbf{h}^\alpha$  for each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ .
- Construct a ReLU FNN to implement a function  $\phi_\alpha : \mathbb{R}^d \rightarrow \mathbb{R}$  approximating  $\partial^\alpha f$  via solving a point fitting problem, i.e.,  $\phi_\alpha$  should fit  $\partial^\alpha f$  well at all points in  $\{\mathbf{x}_\beta : \beta \in \{0, 1, \dots, K-1\}^d\}$  for each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ . That is, for each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ , we need to design  $\phi_\alpha$  to make the following equation true.

$$|\phi_\alpha(\mathbf{x}_\beta) - \partial^\alpha f(\mathbf{x}_\beta)| \leq \mathcal{O}(N^{-2s/d} L^{-2s/d}), \quad \text{for any } \beta \in \{0, 1, \dots, K-1\}^d.$$

We will establish three propositions corresponding to these three steps above. Before showing this construction, we first summarize several propositions as follows. They will be applied to support the construction of the desired ReLU FNNs. Their proofs will be available in the next section.

First, we construct a ReLU FNN  $P_\alpha$  to approximate  $\mathbf{h}^\alpha$  according to Proposition 4.1 below, a general proposition for approximating multivariable polynomials.

**Theorem 4.1.** Assume  $P(\mathbf{x}) = \mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d}$  for  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq k \in \mathbb{N}^+$ . For any  $N, L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $9(N+1) + k - 1$  and depth  $7k^2L$  such that

$$|\phi(\mathbf{x}) - P(\mathbf{x})| \leq 9k(N+1)^{-7kL}, \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

Proposition 4.1 shows that ReLU FNNs with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  is able to approximate polynomials with the rate  $\mathcal{O}(N^{-L})$ . This reveals the power of depth in ReLU FNNs for approximating polynomials, from function compositions. The starting point of a good approximation of functions is to approximate polynomials with high accuracy. In classical approximation theory, approximation power of any numerical scheme depends on the degree of polynomials that can be locally reproduced. Being able to approximate polynomials with high accuracy of deep ReLU FNNs plays a vital role in the proof of Theorem 1.1. It is interesting to study whether there is any other function space with reasonable size, besides polynomial space, having an exponential rate  $\mathcal{O}(N^{-L})$  when approximated by ReLU FNNs. Obviously, the space of smooth function is too big due to the optimality of Theorem 1.1 as shown in Section 2.3.

Proposition 4.1 can be generalized to the case of polynomials defined on an arbitrary hypercube  $[a, b]^d$ . Let us give an example for the polynomial  $xy$  below. Its proof will be provided later in Section 5.

**Lemma 4.2.** For any  $N, L \in \mathbb{N}^+$  and  $a, b \in \mathbb{R}$  with  $a < b$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $9N + 1$  and depth  $L$  such that

$$|\phi(x, y) - xy| \leq 6(b-a)^2 N^{-L}, \quad \text{for any } x, y \in [a, b].$$

Second, we construct a step function  $\Psi$  mapping  $\mathbf{x} \in Q_\beta$  to  $\mathbf{x}_\beta = \frac{\beta}{K}$  for any  $\beta \in \{0, 1, \dots, K-1\}^d$ . We only need to approximate one-dimensional step functions, because in the multidimensional case we can simply set  $\Psi(\mathbf{x}) = [\psi(x_1), \psi(x_2), \dots, \psi(x_d)]^T$ , where  $\psi$  is a one-dimensional step function. In particular, we shall construct ReLU FNNs with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  to approximate step functions with  $\mathcal{O}(K) = \mathcal{O}(N^{2/d} L^{2/d})$  “steps” as in Proposition 4.3 below.

**Proposition 4.3.** For any  $N, L, d \in \mathbb{N}^+$  and  $\delta \in (0, \frac{1}{3K}]$  with  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ , there exists a one-dimensional function  $\phi$  implemented by a ReLU FNN with width  $4\lfloor N^{1/d} \rfloor + 3$  and depth  $4L + 5$  such that

$$\phi(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}\right] \text{ for } k = 0, 1, \dots, K-1.$$

Finally, we construct a ReLU FNN  $\phi_\alpha$  to approximate  $\partial^\alpha f$  via solving a point fitting problem, i.e., we only need  $\phi_\alpha$  to approximate  $\partial^\alpha f$  well at grid points  $\{\mathbf{x}_\beta = \frac{\beta}{K}\}_\beta$  as follows

$$|\phi_\alpha(\mathbf{x}_\beta) - \partial^\alpha f(\mathbf{x}_\beta)| \leq \mathcal{O}(N^{-2s/d} L^{-2s/d}), \quad \text{for any } \beta \in \{0, 1, \dots, K-1\}^d.$$

We can construct ReLU FNNs with width  $\mathcal{O}(sN \ln N)$  and depth  $\mathcal{O}(L \ln L)$  to fit  $\mathcal{O}(N^2 L^2)$  points with an error  $\mathcal{O}(N^{-2s} L^{-2s})$  by Proposition 4.4 below.

**Proposition 4.4.** Given any  $N, L, s \in \mathbb{N}^+$  and  $\xi_i \in [0, 1]$  for  $i = 0, 1, \dots, N^2 L^2 - 1$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $16s(N+1) \log_2(8N)$  and depth  $5(L+2) \log_2(4L)$  such that

(i)  $|\phi(i) - \xi_i| \leq N^{-2s} L^{-2s}$  for  $i = 0, 1, \dots, N^2 L^2 - 1$ ;

(ii)  $0 \leq \phi(x) \leq 1$  for any  $x \in \mathbb{R}$ .

The proofs of Proposition 4.1, 4.3, and 4.4 can be found in Section 5.1, 5.2, and 5.3, respectively. Finally, let us summarize the main ideas of proving Theorem 1.1 in Table 2.

Table 2: A list of sub-networks for approximating smooth functions. Recall that  $\mathbf{h} = \mathbf{x} - \Psi(\mathbf{x}) = \mathbf{x} - \mathbf{x}_\beta$  for  $\mathbf{x} \in Q_\beta$ .

target function	function implemented by network	width	depth	approximation error
step function	$\Psi(\mathbf{x})$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	no error outside $\Omega([0, 1]^d, K, \delta)$
$x_1 x_2$	$\varphi(x_1, x_2)$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{E}_1 = 216(N+1)^{-2s(L+1)}$
$\mathbf{h}^\alpha$	$P_\alpha(\mathbf{h})$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{E}_2 = 9s(N+1)^{-7sL}$
$\partial^\alpha f(\Psi(\mathbf{x}))$	$\phi_\alpha(\Psi(\mathbf{x}))$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{E}_3 = 2N^{-2s} L^{-2s}$
$\sum_{\ \alpha\  \leq s-1} \frac{\partial^\alpha f(\Psi(\mathbf{x}))}{\alpha!} \mathbf{h}^\alpha$	$\sum_{\ \alpha\  \leq s-1} \varphi\left(\frac{\phi_\alpha(\Psi(\mathbf{x}))}{\alpha!}, P_\alpha(\mathbf{h})\right)$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{O}(\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3)$
$f(\mathbf{x})$	$\phi(\mathbf{x}) := \sum_{\ \alpha\  \leq s-1} \varphi\left(\frac{\phi_\alpha(\Psi(\mathbf{x}))}{\alpha!}, P_\alpha(\mathbf{x} - \Psi(\mathbf{x}))\right)$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{O}(\ \mathbf{h}\ _2^{-s} + \mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3) \leq \mathcal{O}(K^{-s}) = \mathcal{O}(N^{-2s/d} L^{-2s/d})$

756

757 Finally, we would like to compare our analysis with that in [45]. Both [45] and our  
758 analysis rely on local Taylor expansions as in (4.1) to approximate the target function  
759  $f(\mathbf{x})$ . Both analysis methods construct ReLU FNNs to approximate polynomials and  
760 encode the Taylor expansion coefficients into ReLU FNNs. However, the way to localize  
761 the Taylor expansion (i.e., defining the local neighborhood such that the expansion is  
762 valid) and the approach to construct ReLU FNNs are different. We will discuss the  
763 details as follows.

764 **Localization.** In [45], a complicated “two-scale” partition procedure and a stan-  
765 dard triangulation divides  $\mathbb{R}^d$  into simplexes and a partition of unity is constructed using  
766 compactly supported functions that are linear on each simplex, which implies that these  
767 functions in the partition of unity can be represented by ReLU FNNs. Taylor expansions  
768 of  $f(\mathbf{x})$  are constructed within each support of the functions in the partition of unity.  
769 In this paper, we simply divide the domain into small hypercubes of uniform size as  
770 visualized in Figure 6. Taylor expansions of  $f(\mathbf{x})$  are constructed within each hyper-  
771 cubes. The reader can understand our approach as a simple way to construct a partition  
772 of unity using piecewise constant functions with binary values. The introduction of the  
773 trifling region allows us to simply construct ReLU FNNs to approximate these piecewise  
774 constant functions without caring about the approximation error within the trifling re-  
775 gion. Hence, our construction can be much simplified and makes it easy to estimate all  
776 constant prefactors in our error estimates, which is challenging in [45].

777 **ReLU FNNs for Taylor expansions.** In [45], very deep ReLU FNNs with width  
778  $\mathcal{O}(1)$  are constructed to approximate polynomials in local Taylor expansions and, hence,  
779 the optimal approximation rate in width was not explored in [45]. In this paper, we con-  
780 struct ReLU FNNs with arbitrary width and depth to approximate polynomials in local  
781 Taylor expansions using Theorem 4.1, which allows us to explore the optimal approxi-  
782 mation rate in width and is more challenging. In [45], the coefficients of adjacent local  
783 Taylor expansions, i.e.,  $\partial^\alpha f$  in (4.1), are encoded into ReLU FNNs via bit extraction,

which is the key to achieve a better approximation rate of ReLU FNNs to approximate  $f(\mathbf{x})$  than the original local Taylor expansions, since the number of coefficients can be significantly reduced via encoding. Actually, the rate in depth by bit extraction is nearly optimal. In this paper, the approximation to  $\partial^\alpha f$  is reduced to a point fitting problem that can be solved by constructing ReLU FNNs using bit extraction as sketched out in the previous paragraphs. Hence, we can also achieve the optimal approximation rate in depth. The key to achieve the optimal approximation rate in width in the above approximation is the application of Lemma 5.4 that essentially fits  $\mathcal{O}(N^2)$  samples with ReLU FNNs of width  $\mathcal{O}(N)$  and depth 2. Due to the simplicity of our analysis, we can construct ReLU FNNs with arbitrary width and depth to approximate  $f(\mathbf{x})$  and specify all constant prefactors in our approximation rate.

## 4.2 Constructive proof

According to the key ideas of proving Theorem 2.2 we summarized in the previous sub-section, we are ready to present the detailed proof.

*Proof of Theorem 2.2.* The detailed proof can be divided into four steps as follows.

**Step 1:** Set up.

Set  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$  and let  $\Omega([0, 1]^d, K, \delta)$  partition  $[0, 1]^d$  into  $K^d$  cubes  $Q_\beta$  for  $\beta \in \{0, 1, \dots, K-1\}^d$ . In particular, for each  $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T \in \{0, 1, \dots, K-1\}^d$ , we define  $\mathbf{x}_\beta := \beta/K$  and

$$Q_\beta := \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T : x_i \in [\frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \leq K-2\}}] \text{ for } i = 1, 2, \dots, d\}.$$

Clearly,  $[0, 1]^d = \Omega([0, 1]^d, K, \delta) \cup (\cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta)$  and  $\mathbf{x}_\beta$  is the vertex of  $Q_\beta$  with minimum  $\|\cdot\|_1$  norm. See Figure 6 for the illustrations of  $Q_\beta$  and  $\mathbf{x}_\beta$ .

By Proposition 4.3, there exists  $\psi \in \mathcal{NN}$  (width  $\leq 4N+3$ ; depth  $\leq 4N+5$ ) such that

$$\psi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}] \text{ for } k = 0, 1, \dots, K-1.$$

Then for each  $\beta \in \{0, 1, \dots, K-1\}^d$ ,  $\psi(x_i) = \beta_i$  for all  $\mathbf{x} \in Q_\beta$  for  $i = 1, 2, \dots, d$ .

Define

$$\Psi(\mathbf{x}) := [\psi(x_1), \psi(x_2), \dots, \psi(x_d)]^T / K, \quad \text{for any } \mathbf{x} \in [0, 1]^d,$$

then

$$\Psi(\mathbf{x}) = \beta/K = \mathbf{x}_\beta, \quad \text{if } \mathbf{x} \in Q_\beta, \quad \text{for } \beta \in \{0, 1, \dots, K-1\}^d.$$

For any  $\mathbf{x} \in Q_\beta$  and  $\beta \in \{0, 1, \dots, K-1\}^d$ , by the Taylor expansion, there exists  $\xi_{\mathbf{x}} \in (0, 1)$  such that

$$f(\mathbf{x}) = \sum_{\|\alpha\|_1 \leq s-1} \frac{\partial^\alpha f(\Psi(\mathbf{x}))}{\alpha!} \mathbf{h}^\alpha + \sum_{\|\alpha\|_1 = s} \frac{\partial^\alpha f(\Psi(\mathbf{x}) + \xi_{\mathbf{x}} \mathbf{h})}{\alpha!} \mathbf{h}^\alpha, \quad \text{where } \mathbf{h} = \mathbf{x} - \Psi(\mathbf{x}).$$

**Step 2:** Construct the desired function  $\phi$ .

817 By Lemma 4.2, there exists  $\varphi \in \mathcal{NN}(\text{width} \leq 9(N+1)+1; \text{depth} \leq 2s(L+1))$  such  
 818 that

$$819 |\varphi(x_1, x_2) - x_1 x_2| \leq 216(N+1)^{-2s(L+1)} =: \mathcal{E}_1, \quad \text{for any } x_1, x_2 \in [-3, 3]. \quad (4.2)$$

820 For each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s$ , by Proposition 4.1, there exists  $P_\alpha$  in

$$821 \mathcal{NN}(\text{width} \leq 9(N+1)+s-1; \text{depth} \leq 7s^2L)$$

822 such that

$$823 |P_\alpha(\mathbf{x}) - \mathbf{x}^\alpha| \leq 9s(N+1)^{-7sL} =: \mathcal{E}_2, \quad \text{for any } \mathbf{x} \in [0, 1]^d. \quad (4.3)$$

824 For each  $i = 0, 1, \dots, K^d - 1$ , define

$$825 \boldsymbol{\eta}(i) = [\eta_1, \eta_2, \dots, \eta_d]^T \in \{0, 1, \dots, K-1\}^d$$

826 such that  $\sum_{j=1}^d \eta_j K^{j-1} = i$ . Such a map  $\boldsymbol{\eta}$  is a bijection from  $\{0, 1, \dots, K^d - 1\}$  to  $\{0, 1, \dots, K-$   
 827  $1\}^d$ . For each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ , define

$$828 \xi_{\alpha, i} = (\partial^\alpha f(\frac{\boldsymbol{\eta}(i)}{K}) + 1)/2, \quad \text{for } i \in \{0, 1, \dots, K^d - 1\}.$$

829 Note that  $K^d = (\lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor)^d \leq N^2 L^2$  and  $\xi_{\alpha, i} \in [0, 1]$  for  $i = 0, 1, \dots, K^d - 1$ . By  
 830 Proposition 4.4, there exists

$$831 \tilde{\phi}_\alpha \in \mathcal{NN}(\text{width} \leq 16s(N+1)\log_2(8N); \text{depth} \leq 5(L+2)\log_2(4L))$$

832 such that, for each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ , we have

$$833 |\tilde{\phi}_\alpha(i) - \xi_{\alpha, i}| \leq N^{-2s} L^{-2s}, \quad \text{for } i = 0, 1, \dots, K^d - 1.$$

834 For each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ , define

$$835 \phi_\alpha(\mathbf{x}) := 2\tilde{\phi}_\alpha\left(\sum_{j=1}^d x_j K^{j-1}\right) - 1, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d.$$

836 It is easy to verify that

$$837 \phi_\alpha \in \mathcal{NN}(\text{width} \leq 16s(N+1)\log_2(8N); \text{depth} \leq 5(L+2)\log_2(4L)).$$

838 Then, for each  $\boldsymbol{\eta} = \boldsymbol{\eta}(i) = [\eta_1, \eta_2, \dots, \eta_d]^T \in \{0, 1, \dots, K-1\}^d$  corresponding to  $i = \sum_{j=1}^d \eta_j K^{j-1} \in$   
 839  $\{0, 1, \dots, K^d - 1\}$ , each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ , we have

$$840 \begin{aligned} \left| \phi_\alpha\left(\frac{\boldsymbol{\eta}}{K}\right) - \partial^\alpha f\left(\frac{\boldsymbol{\eta}}{K}\right) \right| &= \left| 2\tilde{\phi}_\alpha\left(\sum_{j=1}^d \eta_j K^{j-1}\right) - 1 - (2\xi_{\alpha, i} - 1) \right| \\ &= 2|\tilde{\phi}_\alpha(i) - \xi_{\alpha, i}| \leq 2N^{-2s} L^{-2s}. \end{aligned}$$

841 Then, for each  $\beta \in \{0, 1, \dots, K-1\}^d$ , we have

$$842 \left| \phi_\alpha(\mathbf{x}_\beta) - \partial^\alpha f(\mathbf{x}_\beta) \right| = \left| \phi_\alpha\left(\frac{\beta}{K}\right) - \partial^\alpha f\left(\frac{\beta}{K}\right) \right| \leq 2N^{-2s} L^{-2s} =: \mathcal{E}_3. \quad (4.4)$$

Now we can construct the target function  $\phi$  as

$$\phi(\mathbf{x}) := \sum_{\|\alpha\|_1 \leq s-1} \varphi\left(\frac{\phi_\alpha(\Psi(\mathbf{x}))}{\alpha!}, P_\alpha(\mathbf{x} - \Psi(\mathbf{x}))\right), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d. \quad (4.5)$$

It remains to estimate the approximation error and determine the size of the network implementing  $\phi$ .

**Step 3:** Estimate approximation error.

**Fix**  $\beta \in \{0, 1, \dots, K-1\}^d$ , let us estimate the approximation error for a **fixed**  $\mathbf{x} \in Q_\beta$ . See Table 2 for a summary of the approximations errors. Recall that  $\Psi(\mathbf{x}) = \mathbf{x}_\beta$  and  $\mathbf{h} = \mathbf{x} - \Psi(\mathbf{x}) = \mathbf{x} - \mathbf{x}_\beta$ . It is easy to check that  $|f(\mathbf{x}) - \phi(\mathbf{x})|$  is bounded by

$$\begin{aligned} & \left| \sum_{\|\alpha\|_1 \leq s-1} \frac{\partial^\alpha f(\Psi(\mathbf{x}))}{\alpha!} \mathbf{h}^\alpha + \sum_{\|\alpha\|_1 = s} \frac{\partial^\alpha f(\Psi(\mathbf{x}) + \xi_{\mathbf{x}} \mathbf{h})}{\alpha!} \mathbf{h}^\alpha - \sum_{\|\alpha\|_1 \leq s-1} \varphi\left(\frac{\phi_\alpha(\Psi(\mathbf{x}))}{\alpha!}, P_\alpha(\mathbf{x} - \Psi(\mathbf{x}))\right) \right| \\ 851 \quad & \leq \underbrace{\sum_{\|\alpha\|_1 = s} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta + \xi_{\mathbf{x}} \mathbf{h})}{\alpha!} \mathbf{h}^\alpha \right|}_{\mathcal{J}_1} + \underbrace{\sum_{\|\alpha\|_1 \leq s-1} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha - \varphi\left(\frac{\phi_\alpha(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right|}_{\mathcal{J}_2} =: \mathcal{J}_1 + \mathcal{J}_2. \end{aligned}$$

Recall the fact

$$\sum_{\|\alpha\|_1 = s} 1 = |\{\alpha \in \mathbb{N}^d : \|\alpha\|_1 = s\}| \leq (s+1)^{d-1} \quad \textcircled{7}$$

and

$$\sum_{\|\alpha\|_1 \leq s-1} 1 = \sum_{i=0}^{s-1} \left( \sum_{\|\alpha\|_1 = i} 1 \right) \leq \sum_{i=0}^{s-1} (i+1)^{d-1} \leq s \cdot (s-1+1)^{d-1} = s^d.$$

For the first part  $\mathcal{J}_1$ , we have

$$\mathcal{J}_1 = \sum_{\|\alpha\|_1 = s} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta + \xi_{\mathbf{x}} \mathbf{h})}{\alpha!} \mathbf{h}^\alpha \right| \leq \sum_{\|\alpha\|_1 = s} \left| \frac{1}{\alpha!} \mathbf{h}^\alpha \right| \leq (s+1)^{d-1} K^{-s}.$$

Now let us estimate the second part  $\mathcal{J}_2$  as follows.

$$\begin{aligned} \mathcal{J}_2 &= \sum_{\|\alpha\|_1 \leq s-1} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha - \varphi\left(\frac{\phi_\alpha(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right| \\ 859 \quad & \leq \underbrace{\sum_{\|\alpha\|_1 \leq s-1} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha - \varphi\left(\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right|}_{\mathcal{J}_{2,1}} + \underbrace{\sum_{\|\alpha\|_1 \leq s-1} \left| \varphi\left(\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) - \varphi\left(\frac{\phi_\alpha(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right|}_{\mathcal{J}_{2,2}} \\ &=: \mathcal{J}_{2,1} + \mathcal{J}_{2,2}. \end{aligned}$$

Note that  $\mathcal{E}_2 = 9s(N+1)^{-7sL} \leq 9s(2)^{-7s} \leq 2$ . By Equation (4.3) and  $\mathbf{x}^\alpha \in [0, 1]$  for any  $\mathbf{x} \in [0, 1]^d$ , we have  $P_\alpha(\mathbf{x}) \in [-2, 3] \subseteq [-3, 3]$  for any  $\mathbf{x} \in [0, 1]^d$  and  $\alpha \in \mathbb{N}^d$  with

<sup>⑦</sup>In fact, we have  $|\{\alpha \in \mathbb{N}^d : \|\alpha\|_1 = s\}| = \binom{s+d-1}{d-1}$ , implying  $(s/d+1)^{d-1} \leq \sum_{\|\alpha\|_1 = s} 1 \leq (s+1)^{d-1}$ . Thus, the lower bound of the estimate is still exponentially large in  $d$ . To the best of our knowledge, we cannot avoid a constant prefactor that is exponentially large in  $d$  when Taylor expansion is used in the analysis.

862  $\|\alpha\|_1 \leq s-1$ . Then by Equation (4.2) and (4.3), we have, for any  $\mathbf{x} \in Q_\beta$ ,

$$\begin{aligned}
863 \quad \mathcal{J}_{2,1} &= \sum_{\|\alpha\|_1 \leq s-1} \left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha - \varphi\left(\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right| \\
&\leq \sum_{\|\alpha\|_1 \leq s-1} \left( \underbrace{\left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \mathbf{h}^\alpha - \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} P_\alpha(\mathbf{h}) \right|}_{\leq \mathcal{E}_1 \text{ by Eq. (4.2)}} + \underbrace{\left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} P_\alpha(\mathbf{h}) - \varphi\left(\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right|}_{\leq \mathcal{E}_2 \text{ by Eq. (4.3)}} \right) \\
&\leq \sum_{\|\alpha\|_1 \leq s-1} \left( \underbrace{\left( \frac{1}{\alpha!} |\mathbf{h}^\alpha - P_\alpha(\mathbf{h})| \right)}_{\leq \mathcal{E}_1} + \mathcal{E}_1 \right) \leq \sum_{\|\alpha\|_1 \leq s-1} \left( \frac{1}{\alpha!} \mathcal{E}_2 + \mathcal{E}_1 \right) \leq s^d (\mathcal{E}_1 + \mathcal{E}_2).
\end{aligned}$$

864 To estimate  $\mathcal{J}_{2,2}$ , we need the following fact derived from Equation (4.2):

$$\begin{aligned}
865 \quad |\varphi(x_1, x_2) - \varphi(\tilde{x}_1, x_2)| &\leq \underbrace{|\varphi(x_1, x_2) - x_1 x_2|}_{\leq \mathcal{E}_1 \text{ by Eq. (4.2)}} + \underbrace{|\varphi(\tilde{x}_1, x_2) - \tilde{x}_1 x_2|}_{\leq \mathcal{E}_1 \text{ by Eq. (4.2)}} + |x_1 x_2 - \tilde{x}_1 x_2| \\
&\leq 2\mathcal{E}_1 + 3|x_1 - \tilde{x}_1|,
\end{aligned}$$

866 for any  $x_1, \tilde{x}_1, x_2 \in [-3, 3]$ .

867 Since  $\mathcal{E}_3 = 2N^{-2s}L^{-2s} \leq 2$  and  $\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} \in [-1, 1]$  for all  $\mathbf{x} \in Q_\beta$  and each  $\alpha \in \mathbb{N}^d$   
868 with  $\|\alpha\|_1 \leq s-1$ , we have  $\phi_\alpha(\mathbf{x}_\beta) \in [-3, 3]$  by Equation (4.4). By  $P_\alpha(\mathbf{x}) \in [-3, 3]$  and  
869 Equation (4.2) and (4.4), we have, for any  $\mathbf{x} \in Q_\beta$ ,

$$\begin{aligned}
870 \quad \mathcal{J}_{2,2} &= \sum_{\|\alpha\|_1 \leq s-1} \left| \varphi\left(\frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) - \varphi\left(\frac{\phi_\alpha(\mathbf{x}_\beta)}{\alpha!}, P_\alpha(\mathbf{h})\right) \right| \\
&\leq \sum_{\|\alpha\|_1 \leq s-1} \left( 2\mathcal{E}_1 + 3 \underbrace{\left| \frac{\partial^\alpha f(\mathbf{x}_\beta)}{\alpha!} - \frac{\phi_\alpha(\mathbf{x}_\beta)}{\alpha!} \right|}_{\leq \mathcal{E}_3 \text{ by Eq. (4.4)}} \right) \leq \sum_{\|\alpha\|_1 \leq s-1} (2\mathcal{E}_1 + 3\mathcal{E}_3) \leq s^d (2\mathcal{E}_1 + 3\mathcal{E}_3).
\end{aligned}$$

871 Therefore, for any  $\mathbf{x} \in Q_\beta$ ,

$$\begin{aligned}
872 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq \mathcal{J}_1 + \mathcal{J}_2 \leq \mathcal{J}_1 + \mathcal{J}_{2,1} + \mathcal{J}_{2,2} \\
&\leq (s+1)^{d-1} K^{-s} + s^d (\mathcal{E}_1 + \mathcal{E}_2) + s^d (2\mathcal{E}_1 + 3\mathcal{E}_3) \\
&\leq (s+1)^d (K^{-s} + 3\mathcal{E}_1 + \mathcal{E}_2 + 3\mathcal{E}_3).
\end{aligned}$$

873 Since  $\beta \in \{0, 1, \dots, K-1\}^d$  is arbitrary and the fact

$$874 \quad [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta) \subseteq \cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta,$$

875 we have, for any  $\mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ ,

$$876 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq (s+1)^d (K^{-s} + 3\mathcal{E}_1 + \mathcal{E}_2 + 3\mathcal{E}_3).$$

877 Recall that

$$878 \quad (N+1)^{-7sL} \leq (N+1)^{-2s(L+1)} \leq (N+1)^{-2s} 2^{-2sL} \leq N^{-2s} L^{-2s}$$



879 and  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor \geq \frac{N^{2/d} L^{2/d}}{8}$ . Then we have

$$\begin{aligned}
& (s+1)^d (K^{-s} + 3\mathcal{E}_1 + \mathcal{E}_2 + 3\mathcal{E}_3) \\
&= (s+1)^d \left( K^{-s} + 648(N+1)^{-2s(L+1)} + 9s(N+1)^{-7sL} + 6N^{-2s}L^{-2s} \right) \\
&\leq (s+1)^d \left( 8^s N^{-2s/d} L^{-2s/d} + (654 + 9s)N^{-2s}L^{-2s} \right) \\
&\leq (s+1)^d (8^s + 654 + 9s)N^{-2s/d}L^{-2s/d} \leq 84(s+1)^d 8^s N^{-2s/d}L^{-2s/d}.
\end{aligned}$$

880

881 **Step 4:** Determine the size of the network implementing  $\phi$ .

882 It remains to estimate the width and depth of the network implementing  $\phi$ . Recall  
883 that, for  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\|_1 \leq s-1$ ,

$$\begin{cases} \Psi \in \mathcal{NN}(\text{width} \leq d(4N+3); \text{depth} \leq 4L+5), \\ \phi_\alpha \in \mathcal{NN}(\text{width} \leq 16s(N+1)\log_2(8N); \text{depth} \leq 5(L+2)\log_2(4L)), \\ P_\alpha \in \mathcal{NN}(\text{width} \leq 9(N+1)+s-1; \text{depth} \leq 7s^2L), \\ \varphi \in \mathcal{NN}(\text{width} \leq 9N+10; \text{depth} \leq 2s(L+1)). \end{cases}$$

884

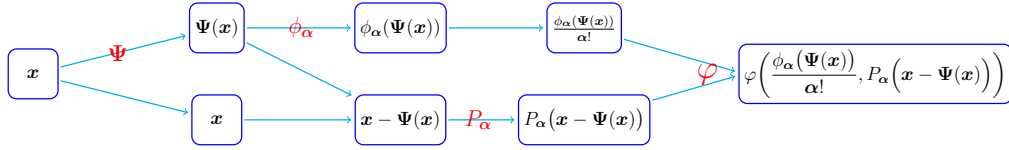


Figure 7: An illustration of the sub-network architecture implementing  $\varphi\left(\frac{\phi_\alpha(\Psi(x))}{\alpha!}, P_\alpha(x - \Psi(x))\right)$  for each  $\alpha \in \mathbb{N}^d$  with  $\|\alpha\| \leq s-1$ .

885 By Equation (4.5) and Figure 7, it easy to verify  $\phi$  can be implemented by a ReLU  
886 FNN with width

$$\begin{aligned}
& \sum_{\|\alpha\|_1 \leq s-1} 16sd(N+2)\log_2(8N) \leq s^d \cdot 16sd(N+2)\log_2(8N) \\
&= 16s^{d+1}d(N+2)\log_2(8N)
\end{aligned}$$

887

888 and depth

$$(4L+5) + 2s(L+1) + 7s^2L + 5(L+2)\log_2(4L) + 3 \leq 18s^2(L+2)\log_2(4L)$$

889

890 as desired. So we finish the proof. □

## 891 5 Proofs of Propositions in Section 4.1

892 In this section, we will prove all propositions in Section 4.1.

## 5.1 Proof of Proposition 4.1 for polynomial approximation

To prove Proposition 4.1, we will construct ReLU FNNs to approximate polynomials following the four steps below.

- $f(x) = x^2$ . We approximate  $f(x) = x^2$  by the combinations and compositions of “sawtooth” functions as shown in Figure 8 and 9.
- $f(x, y) = xy$ . To approximate  $f(x, y) = xy$ , we use the result of the previous step and the fact  $xy = 2\left(\left(\frac{x+y}{2}\right)^2 - \left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2\right)$ .
- $f(x_1, x_2, \dots, x_k) = x_1 x_2 \dots x_k$ . We approximate  $f(x_1, x_2, \dots, x_k) = x_1 x_2 \dots x_k$  for any  $k \geq 2$  via mathematical induction based on the result of the previous step.
- A general polynomial  $P(\mathbf{x}) = \mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d}$  with  $\|\alpha\|_1 \leq k$ . Any one-term polynomial of degree  $\leq k$  can be written as  $C z_1 z_2 \dots z_k$  with some entries equaling 1, where  $C$  is a constant and  $\mathbf{z} = [z_1, z_2, \dots, z_k]^T$  can be attained via a linear map with  $\mathbf{x}$  as the input. Then use the result of the previous step.

The idea of using “sawtooth” functions (see Figure 8) was first raised in [43] for approximating  $x^2$  using FNNs with width 6 and depth  $\mathcal{O}(L)$  and achieving an error  $\mathcal{O}(2^{-L})$ ; our construction is different to and more general than that in [43], working for ReLU FNNs of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  for any  $N$  and  $L$ , and achieving an error  $\mathcal{O}(N^{-L})$ . As discussed above below Proposition 4.1, this  $\mathcal{O}(N^{-L})$  approximation rate of polynomial functions shows the power of depth in ReLU FNNs via function composition.

First, let us show how to construct ReLU FNNs to approximate  $f(x) = x^2$ .

**Lemma 5.1.** *For any  $N, L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $3N$  and depth  $L$  such that*

$$|\phi(x) - x^2| \leq N^{-L}, \quad \text{for any } x \in [0, 1].$$

*Proof.* Define a set of “sawtooth” functions  $T_i : [0, 1] \rightarrow [0, 1]$  by induction as follows. Let

$$T_1(x) = \begin{cases} 2x, & x \leq \frac{1}{2}, \\ 2(1-x), & x > \frac{1}{2}, \end{cases}$$

and

$$T_i = T_{i-1} \circ T_1, \quad \text{for } i = 2, 3, \dots.$$

It is easy to check that  $T_i$  has  $2^{i-1}$  “sawtooth” and

$$T_{m+n} = T_m \circ T_n, \quad \text{for any } m, n \in \mathbb{N}^+.$$

See Figure 8 for illustrations of  $T_i$  for  $i = 1, 2, 3, 4$ .

Define piecewise linear functions  $f_s : [0, 1] \rightarrow [0, 1]$  for  $s \in \mathbb{N}^+$  satisfying the following two requirements (see Figure 9 for several examples of  $f_s$ ).

- $f_s\left(\frac{j}{2^s}\right) = \left(\frac{j}{2^s}\right)^2$  for  $j = 0, 1, 2, \dots, 2^s$ .
- $f_s(x)$  is linear between any two adjacent points of  $\{\frac{j}{2^s} : j = 0, 1, 2, \dots, 2^s\}$ .

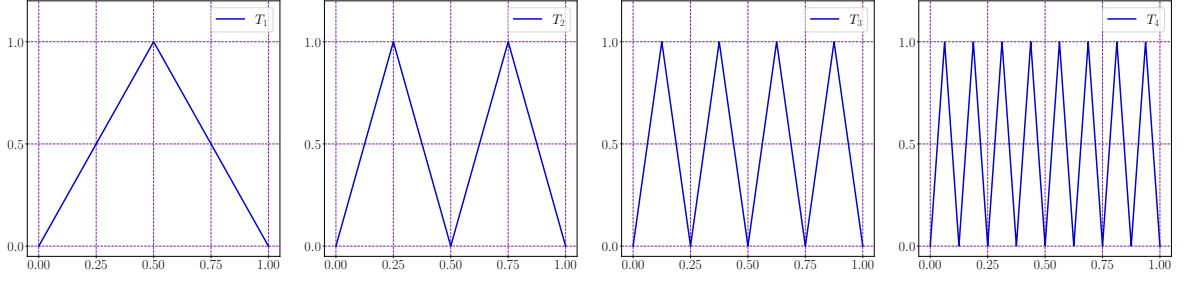


Figure 8: Examples of “sawtooth” functions  $T_1$ ,  $T_2$ ,  $T_3$ , and  $T_4$ .

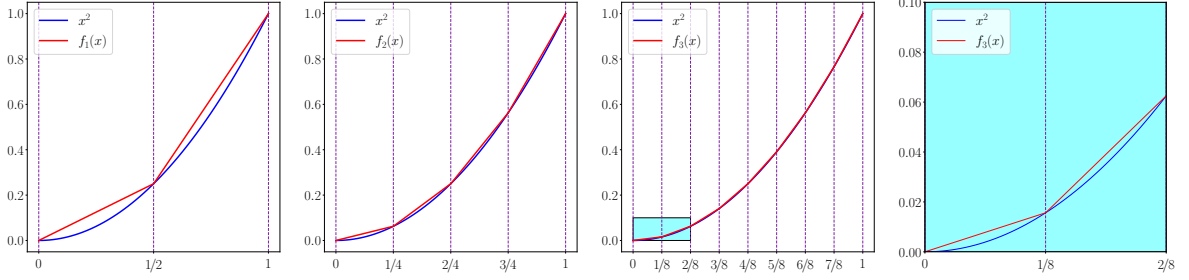


Figure 9: Illustrations of  $f_1$ ,  $f_2$ , and  $f_3$  for approximating  $x^2$ .

928 Recall the fact  $\frac{(x-h)^2+(x+h)^2}{2} - x^2 = h^2$  for any  $h > 0$ . It is easy to check that

$$929 \quad |x^2 - f_s(x)| \leq (2^{-(s+1)})^2 = 2^{-2(s+1)}, \quad \text{for any } x \in [0, 1] \text{ and } s \in \mathbb{N}^+. \quad (5.1)$$

930 Note that  $f_{i-1}(x) = f_i(x) = x^2$  for  $x \in \{\frac{j}{2^{i-1}} : j = 0, 1, 2, \dots, 2^{i-1}\}$  and the graph of  $f_{i-1} - f_i$   
 931 is a symmetric “sawtooth” between any two adjacent points of  $\{\frac{j}{2^{i-1}} : j = 0, 1, 2, \dots, 2^{i-1}\}$ .  
 932 Thus, we have

$$933 \quad f_{i-1}(x) - f_i(x) = \frac{T_i(x)}{2^{2i}}, \quad \text{for any } x \in [0, 1] \text{ and } i = 2, 3, \dots.$$

934 Therefore, for any  $x \in [0, 1]$  and  $s \in \mathbb{N}^+$ , we have

$$935 \quad f_s(x) = f_1(x) + \sum_{i=2}^s (f_i - f_{i-1}) = x - (x - f_1(x)) - \sum_{i=2}^s \frac{T_i(x)}{2^{2i}} = x - \sum_{i=1}^s \frac{T_i(x)}{2^{2i}}.$$

936 Given  $N \in \mathbb{N}^+$ , there exists a unique  $k \in \mathbb{N}^+$  such that  $(k-1)2^{k-1} + 1 \leq N \leq k2^k$ .  
 937 For this  $k$ , using  $s = Lk$ , we can construct a ReLU FNN as shown in Figure 10 to  
 938 implement a function  $\phi = f_{Lk}$  approximating  $x^2$  well. Note that  $T_i$  can be implemented  
 939 by a one-hidden-layer ReLU FNN with width  $2^i$ . Hence, the network in Figure 10 has  
 940 width  $k2^k + 1 \leq 3N$ <sup>⑧</sup> and depth  $2L$ .

941 As shown in Figure 10,  $(2\ell)$ -th hidden layer of the network has the identify function  
 942 as their activation functions for  $\ell = 1, 2, \dots, L$ . Thus, the network in Figure 10 can  
 943 be interpreted as a ReLU FNN with width  $3N$  and depth  $L$ . In fact, if all activation  
 944 functions in a certain hidden layer are identity maps, the depth can be reduced by one via

---

<sup>⑧</sup>This inequality is clear for  $k = 1, 2, 3, 4$ . In the case  $k \geq 5$ , we have  $k2^k + 1 \leq \frac{k2^k+1}{N}N \leq \frac{(k+1)2^k}{(k-1)2^{k-1}}N \leq 2\frac{k+1}{k-1}N \leq 3N$ .

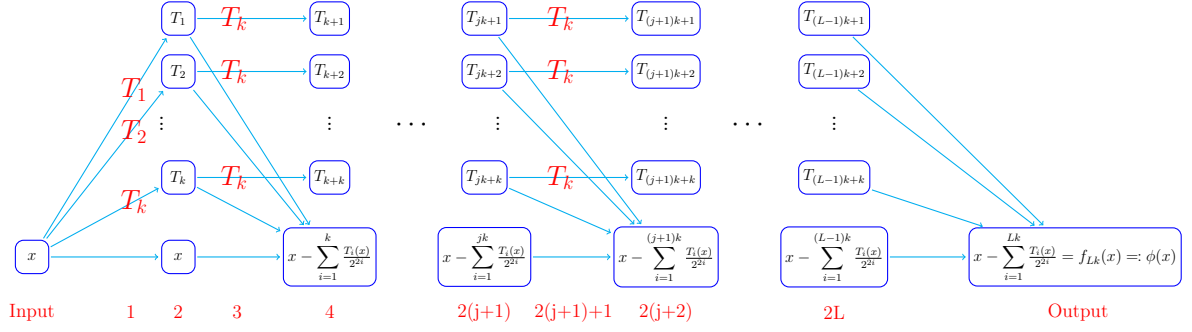


Figure 10: An illustration of the target network architecture for approximating  $x^2$  on  $[0,1]$ .  $T_i$  can be implemented by a one-hidden-layer ReLU FNN with width  $2^i$  for  $i = 1, 2, \dots, K$ . The red numbers below the architecture indicate the order of hidden layers.

combining adjacent two linear transforms into one. For example, suppose  $\mathbf{W}_1 \in \mathbb{R}^{N_1 \times N_2}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{N_2 \times N_3}$ , and  $\varrho$  is an identity map that can be applied to vectors or matrices elementwisely, then  $\mathbf{W}_1 \varrho(\mathbf{W}_2 \mathbf{x}) = \mathbf{W}_3 \mathbf{x}$  for any  $\mathbf{x} \in \mathbb{R}^{N_3}$ , where  $\mathbf{W}_3 = \mathbf{W}_1 \cdot \mathbf{W}_2 \in \mathbb{R}^{N_1 \times N_3}$ .

It remains to estimate the approximation error of  $\phi(x) \approx x^2$ . By Equation (5.1), for any  $x \in [0, 1]$ , we have

$$|x^2 - \phi(x)| = |x^2 - f_{Lk}(x)| \leq 2^{-2(Lk+1)} \leq 2^{-2Lk} \leq N^{-L},$$

where the last inequality comes from  $N \leq k2^k \leq 2^{2k}$ . So we finish the proof.  $\square$

We have constructed a ReLU FNN to approximate  $f(x) = x^2$ . By the fact  $xy = 2\left(\left(\frac{x+y}{2}\right)^2 - \left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2\right)$ , it is easy to construct a new ReLU FNN to approximate  $f(x, y) = xy$  as follows.

**Lemma 5.2.** *For any  $N, L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $9N$  and depth  $L$  such that*

$$|\phi(x, y) - xy| \leq 6N^{-L}, \quad \text{for any } x, y \in [0, 1].$$

*Proof.* By Lemma 5.1, there exists a function  $\psi$  implemented by a ReLU FNN with width  $3N$  and depth  $L$  such that

$$|x^2 - \psi(x)| \leq N^{-L}, \quad \text{for any } x \in [0, 1].$$

Together with the fact

$$xy = 2\left(\left(\frac{x+y}{2}\right)^2 - \left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2\right), \quad \text{for any } x, y \in \mathbb{R},$$

we construct the target function  $\phi$  as

$$\phi(x, y) := 2\left(\psi\left(\frac{x+y}{2}\right) - \psi\left(\frac{x}{2}\right) - \psi\left(\frac{y}{2}\right)\right), \quad \text{for any } x, y \in \mathbb{R}. \quad (5.2)$$

Then  $\phi$  can be implemented by the network architecture in Figure 11.

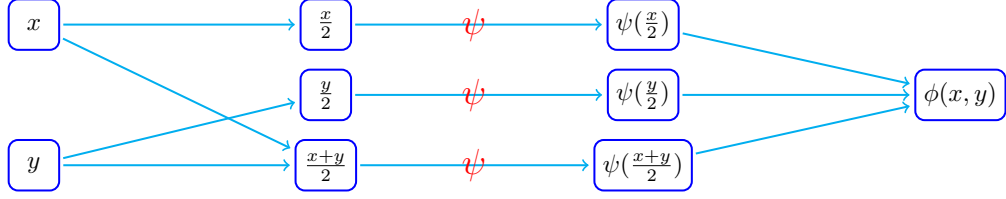


Figure 11: An illustration of the network architecture implementing  $\phi$  for approximating  $xy$  on  $[0, 1]^2$ .

It follows from  $\psi \in \mathcal{NN}(\text{width} \leq 3N; \text{depth} \leq L)$  that the network in Figure 11 is with width  $9N$  and depth  $L + 2$ . Similar to the discussion in the proof of Lemma 5.1, the network in Figure 11 can be interpreted as a ReLU FNN with width  $9N$  and depth  $L$ , since two of hidden layers has the identify function as their activation functions. Moreover, for any  $x, y \in [0, 1]$ ,

$$|xy - \phi(x, y)| = \left| 2\left(\left(\frac{x+y}{2}\right)^2 - \left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2\right) - 2\left(\psi\left(\frac{x+y}{2}\right) - \psi\left(\frac{x}{2}\right) - \psi\left(\frac{y}{2}\right)\right) \right|$$

$$\leq 2\left|\left(\frac{x+y}{2}\right)^2 - \psi\left(\frac{x+y}{2}\right)\right| + 2\left|\left(\frac{x}{2}\right)^2 - \psi\left(\frac{x}{2}\right)\right| + 2\left|\left(\frac{y}{2}\right)^2 - \psi\left(\frac{y}{2}\right)\right| \leq 6N^{-L}.$$

Therefore, we have finished the proof.  $\square$

Now let us prove Lemma 4.2 that shows how to construct a ReLU FNN to approximate  $f(x, y) = xy$  on  $[a, b]^2$  with arbitrary  $a < b$ , i.e., a rescaled version of Lemma 5.2.

*Proof of Lemma 4.2.* By Lemma 5.2, there exists a function  $\psi$  implemented by a ReLU FNN with width  $9N$  and depth  $L$  such that

$$|\psi(\tilde{x}, \tilde{y}) - \tilde{x}\tilde{y}| \leq 6N^{-L}, \quad \text{for any } \tilde{x}, \tilde{y} \in [0, 1].$$

By setting  $\tilde{x} = \frac{x-a}{b-a}$  and  $\tilde{y} = \frac{y-a}{b-a}$  for any  $x, y \in [a, b]$ , we have  $\tilde{x}, \tilde{y} \in [0, 1]$ , implying

$$\left| \psi\left(\frac{x-a}{b-a}, \frac{y-a}{b-a}\right) - \frac{x-a}{b-a} \frac{y-a}{b-a} \right| \leq 6N^{-L}, \quad \text{for any } x, y \in [a, b].$$

It follows that, for any  $x, y \in [a, b]$ ,

$$\left| (b-a)^2 \psi\left(\frac{x-a}{b-a}, \frac{y-a}{b-a}\right) + a(x+y) - a^2 - xy \right| \leq 6(b-a)^2 N^{-L}.$$

Define, for any  $x, y \in \mathbb{R}$ ,

$$\phi(x, y) := (b-a)^2 \psi\left(\frac{x-a}{b-a}, \frac{y-a}{b-a}\right) + a \cdot \sigma(x+y+2|a|) - a^2 - 2a|a|.$$

Then  $\phi$  can be implemented by the network architecture in Figure 12.

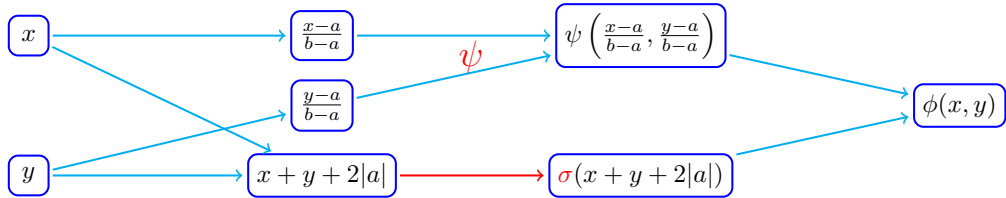


Figure 12: An illustration of the network architecture implementing  $\phi$  for approximating  $xy$  on  $[a, b]^2$ . Two of hidden layers has the identify function as their activation functions, since the red “ $\sigma$ ” comes from the red arrow “ $\rightarrow$ ”, where the red arrow “ $\rightarrow$ ” is a ReLU FNN with width 1 and depth  $L$ .

If follows from  $\psi \in \mathcal{NN}(\text{width} \leq 9N; \text{depth} \leq L)$  that the network in Figure 12 is with width  $9N + 1$  and depth  $L + 2$ . Similar to the discussion in the proof of Lemma 5.1, the network in Figure 12 can be interpreted as a ReLU FNN with width  $9N + 1$  and depth  $L$ , since two of hidden layers has the identify function as their activation functions.

Note that  $x + y + 2|a| \geq 0$  for any  $x, y \in [a, b]$ , implying

$$\phi(x, y) = (b - a)^2 \psi\left(\frac{x-a}{b-a}, \frac{y-a}{b-a}\right) + a(x + y) - a^2, \quad \text{for any } x, y \in [a, b].$$

Hence,

$$|\phi(x, y) - xy| \leq 6(b - a)^2 N^{-L}, \quad \text{for any } x, y \in [a, b].$$

So we finish the proof.  $\square$

The next lemma constructs a ReLU FNN to approximate a multivariable function  $f(x_1, x_2, \dots, x_k) = x_1 x_2 \dots x_k$  on  $[0, 1]^k$ .

**Lemma 5.3.** *For any  $N, L, k \in \mathbb{N}^+$  with  $k \geq 2$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $9(N + 1) + k - 1$  and depth  $7kL(k - 1)$  such that*

$$|\phi(\mathbf{x}) - x_1 x_2 \dots x_k| \leq 9(k - 1)(N + 1)^{-7kL}, \quad \text{for } \mathbf{x} = (x_1, x_2, \dots, x_k) \in [0, 1]^k.$$

*Proof.* By Lemma 4.2, there exists a function  $\phi_1$  implemented by a ReLU FNN with width  $9(N + 1) + 1$  and depth  $7kL$  such that

$$|\phi_1(x, y) - xy| \leq 6(1.2)^2 (N + 1)^{-7kL} \leq 9(N + 1)^{-7kL}, \quad \text{for } x, y \in [-0.1, 1.1]. \quad (5.3)$$

Next, we construct a sequence of functions  $\phi_i : [0, 1]^{i+1} \rightarrow [0, 1]$  for  $i \in \{1, 2, \dots, k - 1\}$  by induction such that

(i)  $\phi_i$  can be implemented by a ReLU FNN with width  $9(N + 1) + i$  and depth  $7kLi$  for each  $i \in \{1, 2, \dots, k - 1\}$ .

(ii) For any  $i \in \{1, 2, \dots, k - 1\}$  and  $x_1, x_2, \dots, x_{i+1} \in [0, 1]$ , it holds that

$$|\phi_i(x_1, \dots, x_{i+1}) - x_1 x_2 \dots x_{i+1}| \leq 9i(N + 1)^{-7kL}. \quad (5.4)$$

First, let us consider the case  $i = 1$ , it is obvious that the two required conditions are true: 1)  $9(N + 1) + i = 9(N + 1) + 1$  and  $7kLi = 7kL$  if  $i = 1$ ; 2) Equation (5.3) implies Equation (5.4) for  $i = 1$ .

Now assume  $\phi_i$  has been defined, we define

$$\phi_{i+1}(x_1, \dots, x_{i+2}) := \phi_1(\phi_i(x_1, \dots, x_{i+1}), \sigma(x_{i+2})), \quad \text{for any } x_1, \dots, x_{i+2} \in \mathbb{R}.$$

Note that  $\phi_i \in \mathcal{NN}(\text{width} \leq 9(N + 1) + i; \text{depth} \leq 7kLi)$  and  $\phi_1 \in \mathcal{NN}(\text{width} \leq 9(N + 1) + 1; \text{depth} \leq 7kL)$ . Then  $\phi_{i+1}$  can be implemented via a ReLU FNN with width

$$\max\{9(N + 1) + i + 1, 9(N + 1) + 1\} = 9(N + 1) + (i + 1)$$

and depth  $7kLi + 7kL = 7kL(i + 1)$ .

By the hypothesis of induction, we have

$$|\phi_i(x_1, \dots, x_{i+1}) - x_1 x_2 \dots x_{i+1}| \leq 9i(N+1)^{-7kL}. \quad (5.5)$$

Recall the fact  $9i(N+1)^{-7kL} \leq 9k2^{-7k} \leq 9k\frac{2^{-7}}{k} \leq 0.1$  for any  $N, L, k \in \mathbb{N}^+$  and  $i \in \{1, 2, \dots, k-1\}$ . It follows that

$$\phi_i(x_1, \dots, x_{i+1}) \in [-0.1, 1.1], \quad \text{for any } x_1, \dots, x_{i+1} \in [0, 1].$$

Therefore, by Equation (5.3) and (5.5), we have

$$\begin{aligned} & |\phi_{i+1}(x_1, \dots, x_{i+2}) - x_1 x_2 \dots x_{i+2}| \\ &= |\phi_1(\phi_i(x_1, \dots, x_{i+1}), \sigma(x_{i+2})) - x_1 x_2 \dots x_{i+2}| \\ &\leq |\phi_1(\phi_i(x_1, \dots, x_{i+1}), x_{i+2}) - \phi_i(x_1, \dots, x_{i+1})x_{i+2}| + |\phi_i(x_1, \dots, x_{i+1})x_{i+2} - x_1 x_2 \dots x_{i+2}| \\ &\leq 9(N+1)^{-7kL} + 9i(N+1)^{-7kL} = 9(i+1)(N+1)^{-7kL}, \end{aligned}$$

for any  $x_1, x_2, \dots, x_{i+2} \in [0, 1]$ , which means we finish the process of induction.

Now let  $\phi := \phi_{k-1}$ , by the principle of induction, we have

$$|\phi(x_1, \dots, x_k) - x_1 x_2 \dots x_k| \leq 9(k-1)(N+1)^{-7kL}, \quad \text{for any } x_1, \dots, x_k \in [0, 1].$$

So  $\phi$  is the desired function implemented by a ReLU FNN with width  $9(N+1) + k - 1$  and depth  $7kL(k-1)$ , which means we finish the proof.  $\square$

With Lemma 5.3 in hand, we are ready to prove Proposition 4.1 for approximating general multivariable polynomials by ReLU FNNs.

*Proof of Proposition 4.1.* The case  $k = 1$  is trivial, so we assume  $k \geq 2$  below. Set  $\tilde{k} = \|\alpha\|_1 \leq k$ , and denote  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_d]^T$ , and let  $[z_1, z_2, \dots, z_{\tilde{k}}]^T \in \mathbb{R}^{\tilde{k}}$  be the vector such that

$$z_\ell = x_j, \quad \text{if } \sum_{i=1}^{j-1} \alpha_i < \ell \leq \sum_{i=1}^j \alpha_i, \quad \text{for } j = 1, 2, \dots, d.$$

That is,

$$[z_1, z_2, \dots, z_{\tilde{k}}]^T = \left[ \overbrace{x_1, \dots, x_1}^{\alpha_1 \text{ times}}, \overbrace{x_2, \dots, x_2}^{\alpha_2 \text{ times}}, \dots, \overbrace{x_d, \dots, x_d}^{\alpha_d \text{ times}} \right]^T \in \mathbb{R}^{\tilde{k}}.$$

Then we have  $P(\mathbf{x}) = \mathbf{x}^\alpha = z_1 z_2 \dots z_{\tilde{k}}$ .

We construct the target ReLU FNN in two steps. First, there exists an affine linear map  $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}^k$  that duplicates  $\mathbf{x}$  to form a new vector  $[z_1, z_2, \dots, z_{\tilde{k}}, 1, \dots, 1]^T \in \mathbb{R}^k$ , i.e.,  $\mathcal{L}(\mathbf{x}) = [z_1, z_2, \dots, z_{\tilde{k}}, 1, \dots, 1]^T \in \mathbb{R}^k$ . Second, by Lemma 5.3, there exists a function  $\psi : \mathbb{R}^k \rightarrow \mathbb{R}$  implemented by a ReLU FNN with width  $9(N+1) + k - 1$  and depth  $7kL(k-1)$  such that  $\psi$  maps  $[z_1, z_2, \dots, z_{\tilde{k}}, 1, \dots, 1]^T \in \mathbb{R}^k$  to  $z_1 z_2 \dots z_{\tilde{k}}$  within an error  $9(k-1)(N+1)^{-7kL}$ . Hence, we can construct our final target function via  $\phi := \psi \circ \mathcal{L}$ . Then  $\phi$  can be implemented by a ReLU FNN with width  $9(N+1) + k - 1$  and depth  $7kL(k-1) \leq 7k^2L$ , and

$$\begin{aligned} |\phi(\mathbf{x}) - P(\mathbf{x})| &= |\phi(\mathbf{x}) - \mathbf{x}^\alpha| = |\psi \circ \mathcal{L}(\mathbf{x}) - x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d}| \\ &= |\psi(z_1, z_2, \dots, z_{\tilde{k}}, 1, \dots, 1) - z_1 z_2 \dots z_{\tilde{k}}| \\ &\leq 9(k-1)(N+1)^{-7kL} \leq 9k(N+1)^{-7kL}, \end{aligned}$$

for any  $x_1, x_2, \dots, x_d \in [0, 1]$ . So, we finish the proof.  $\square$



## 5.2 Proof of Proposition 4.3 for step function approximation

To prove Proposition 4.3 in this sub-section, we will discuss how to pointwisely approximate step functions by ReLU FNNs except for a trifling region. Before proving Proposition 4.3, let us first introduce a basic lemma about fitting  $\mathcal{O}(N_1 N_2)$  samples using a two-hidden-layer ReLU FNN with  $\mathcal{O}(N_1 + N_2)$  neurons.

**Lemma 5.4.** *For any  $N_1, N_2 \in \mathbb{N}^+$ , given  $N_1(N_2 + 1) + 1$  samples  $(x_i, y_i) \in \mathbb{R}^2$  with  $x_0 < x_1 < \dots < x_{N_1(N_2+1)}$  and  $y_i \geq 0$  for  $i = 0, 1, \dots, N_1(N_2+1)$ , there exists  $\phi \in \mathcal{NN}(\text{\#input} = 1; \text{widthvec} = [2N_1, 2N_2 + 1])$  satisfying the following conditions.*

1.  $\phi(x_i) = y_i$  for  $i = 0, 1, \dots, N_1(N_2 + 1)$ ;
2.  $\phi$  is linear on each interval  $[x_{i-1}, x_i]$  for  $i \notin \{(N_2 + 1)j : j = 1, 2, \dots, N_1\}$ .

The above lemma is Lemma 2.2 of [40] and the reader is referred to [40] for its proof. Essentially, this lemma shows the equivalence of one-hidden-layer ReLU FNNs of size  $\mathcal{O}(N^2)$  and two-hidden-layer ones of size  $\mathcal{O}(N)$  to fit  $\mathcal{O}(N^2)$  samples.

The next lemma below shows that special shallow and wide ReLU FNNs can be represented by deep and narrow ones. This lemma was proposed as Proposition 2.2 in [41].

**Lemma 5.5.** *For any  $N, L, d \in \mathbb{N}^+$ , it holds that*

$$\begin{aligned} & \mathcal{NN}(\text{\#input} = d; \text{widthvec} = [N, NL]; \text{\#output} = 1) \\ & \subseteq \mathcal{NN}(\text{\#input} = d; \text{width} \leq 2N + 2; \text{depth} \leq L + 1; \text{\#output} = 1). \end{aligned}$$

Now, let us present the detailed proof of Proposition 4.3.

*Proof of Proposition 4.3.* We divide the proof into two cases:  $d = 1$  and  $d \geq 2$ .

**Case 1:**  $d = 1$ .

In this case,  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor = N^2 L^2$ . Denote  $M = N^2 L$  and consider the sample set

$$\begin{aligned} & \{(1, M - 1), (2, 0)\} \cup \left\{ \left( \frac{m}{M}, m \right) : m = 0, 1, \dots, M - 1 \right\} \\ & \cup \left\{ \left( \frac{m+1}{M} - \delta, m \right) : m = 0, 1, \dots, M - 2 \right\}. \end{aligned}$$

Its size is  $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$ . By Lemma 5.4 (set  $N_1 = N$  and  $N_2 = 2NL - 1$  therein), there exists

$$\begin{aligned} \phi_1 & \in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL - 1) + 1]) \\ & = \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \end{aligned}$$

such that

- $\phi_1(\frac{M-1}{M}) = \phi_1(1) = M - 1$  and  $\phi_1(\frac{m}{M}) = \phi_1(\frac{m+1}{M} - \delta) = m$  for  $m = 0, 1, \dots, M - 2$ ;
- $\phi_1$  is linear on  $[\frac{M-1}{M}, 1]$  and each interval  $[\frac{m}{M}, \frac{m+1}{M} - \delta]$  for  $m = 0, 1, \dots, M - 2$ .

1078 Then

$$1079 \quad \phi_1(x) = m, \quad \text{if } x \in \left[\frac{m}{M}, \frac{m+1}{M} - \delta \cdot 1_{\{m \leq M-2\}}\right], \quad \text{for } m = 0, 1, \dots, M-1. \quad (5.6)$$

1080 Now consider the another sample set

$$1081 \quad \left\{\left(\frac{1}{M}, L-1\right), (2, 0)\right\} \cup \left\{\left(\frac{\ell}{ML}, \ell\right) : \ell = 0, 1, \dots, L-1\right\} \\ \cup \left\{\left(\frac{\ell+1}{ML} - \delta, \ell\right) : \ell = 0, 1, \dots, L-2\right\}.$$

1082 Its size is  $2L + 1 = 1 \cdot ((2L - 1) + 1) + 1$ . By Lemma 5.4 (set  $N_1 = 1$  and  $N_2 = 2L - 1$   
1083 therein), there exists

$$1084 \quad \phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 2(2L - 1) + 1]) \\ = \mathcal{NN}(\text{widthvec} = [2, 4L - 1])$$

1085 such that

- 1086 •  $\phi_2(\frac{L-1}{ML}) = \phi_2(\frac{1}{M}) = L - 1$  and  $\phi_2(\frac{\ell}{ML}) = \phi_2(\frac{\ell+1}{ML} - \delta) = \ell$  for  $\ell = 0, 1, \dots, L - 2$ ;
- 1087 •  $\phi_2$  is linear on  $[\frac{L-1}{ML}, \frac{1}{M}]$  and each interval  $[\frac{\ell}{ML}, \frac{\ell+1}{ML} - \delta]$  for  $\ell = 0, 1, \dots, L - 2$ .

1088 It follows that, for  $m = 0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ ,

$$1089 \quad \phi_2(x - \frac{m}{M}) = \ell, \quad \text{for } x \in \left[\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{\ell \leq L-2\}}\right]. \quad (5.7)$$

1090  $K = ML$  implies any  $k \in \{0, 1, \dots, K - 1\}$  can be unique represented by  $k = mL + \ell$  for  
1091  $m = 0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ . Then the desired function  $\phi$  can be implemented  
1092 by ReLU FNN shown in Figure 13.

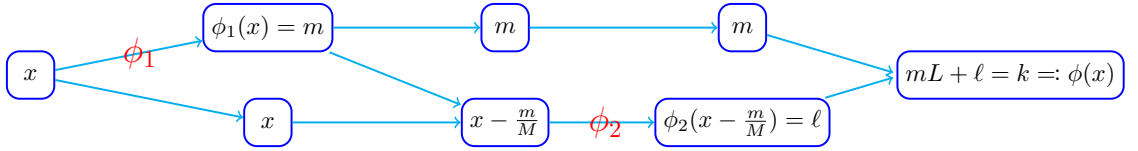


Figure 13: An illustration of the network architecture implementing  $\phi$  based on Equation (5.6) and (5.7) with  $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}] = [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{m \leq M-2 \text{ or } \ell \leq L-2\}}]$ , where  $k = mL + \ell$  for  $m = 0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ .

1093 Clearly,

$$1094 \quad \phi(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}\right] \text{ for } k \in \{0, 1, \dots, K - 1\}.$$

1095 By Lemma 5.5,  $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \subseteq \mathcal{NN}(\text{width} \leq 4N + 2; \text{depth} \leq$   
1096  $2L + 1)$  and  $\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 4L - 1]) \subseteq \mathcal{NN}(\text{width} \leq 6; \text{depth} \leq 2L + 1)$ , implying  
1097  $\phi \in \mathcal{NN}(\text{width} \leq \max\{4N + 2 + 1, 6 + 1\} = 4N + 3; \text{depth} \leq (2L + 1) + 2 + (2L + 1) + 1 = 4L + 5)$ .  
1098 So we finish the proof for the case  $d = 1$

1099 **Case 2:**  $d \geq 2$ .

1100 Now we consider the case when  $d \geq 2$ . Consider the sample set

$$1101 \quad \left\{(1, K - 1), (2, 0)\right\} \cup \left\{\left(\frac{k}{K}, k\right) : k = 0, 1, \dots, K - 1\right\} \\ \cup \left\{\left(\frac{k+1}{K} - \delta, k\right) : k = 0, 1, \dots, K - 2\right\},$$

whose size is  $2K + 1 = \lfloor N^{1/d} \rfloor ((2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1) + 1$ . By Lemma 5.4 (set  $N_1 = \lfloor N^{1/d} \rfloor$  and  $N_2 = 2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1$  therein), there exists

$$\begin{aligned} \phi &\in \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 2(2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1]) \\ &= \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \end{aligned}$$

such that

- $\phi(\frac{K-1}{K}) = \phi(1) = K - 1$ , and  $\phi(\frac{k}{K}) = \phi(\frac{k+1}{K} - \delta) = k$  for  $k = 0, 1, \dots, K - 2$ ;
- $\phi$  is linear on  $[\frac{K-1}{K}, 1]$  and each interval  $[\frac{k}{K}, \frac{k+1}{K} - \delta]$  for  $k = 0, 1, \dots, K - 2$ .

Then

$$\phi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}] \text{ for } k = 0, 1, \dots, K - 1.$$

By Lemma 5.5,

$$\begin{aligned} \phi &\in \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \\ &\subseteq \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 2; \text{depth} \leq 2\lfloor L^{2/d} \rfloor + 1) \\ &\subseteq \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 4L + 5). \end{aligned}$$

which means we finish the proof for the case  $d \geq 2$ . □

### 5.3 Proof of Proposition 4.4 for point fitting

In this sub-section, we will discuss how to use ReLU FNNs to fit a collection of points in  $\mathbb{R}^2$ .<sup>⑨</sup> It is trivial to fit  $n$  points via one-hidden-layer ReLU FNNs with  $\mathcal{O}(n)$  parameters. However, to prove Proposition 4.4, we need to fit  $\mathcal{O}(n)$  points with much less parameters, which is the main difficulty of our proof. Our proof below is mainly based on the “bit extraction” technique and the composition architecture of neural networks.

Let us first introduce a basic lemma based on the “bit extraction” technique, which is in fact Lemma 2.6 of [41].

**Lemma 5.6.** *For any  $N, L \in \mathbb{N}^+$ , any  $\theta_{m,\ell} \in \{0, 1\}$  for  $m = 0, 1, \dots, M-1$  and  $\ell = 0, 1, \dots, L-1$ , where  $M = N^2L$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $4N + 3$  and depth  $3L + 3$  such that*

$$\phi(m, \ell) = \sum_{j=0}^{\ell} \theta_{m,j}, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1.$$

Next, let us introduce Lemma 5.7, a variant of Lemma 5.6 for a different mapping for the “bit extraction”. Its proof is based on Lemma 5.4, 5.5, and 5.6.

**Lemma 5.7.** *For any  $N, L \in \mathbb{N}^+$  and any  $\theta_i \in \{0, 1\}$  for  $i = 0, 1, \dots, N^2L^2 - 1$ , there exists a function  $\phi$  implemented by a ReLU FNN with width  $8N + 6$  and depth  $5L + 7$  such that*

$$\phi(i) = \theta_i, \quad \text{for } i = 0, 1, \dots, N^2L^2 - 1.$$

---

<sup>⑨</sup>Fitting a collection of points  $\{(x_i, y_i)\}_i$  in  $\mathbb{R}^2$  means that the target ReLU FNN takes a value close to  $y_i$  at the location  $x_i$ .

1130 *Proof.* The case  $L = 1$  is clear. We assume  $L \geq 2$  below.

1131 Denote  $M = N^2L$ , for each  $i \in \{0, 1, \dots, N^2L - 1\}$ , there exists a unique representation  
 1132  $i = mL + \ell$  for  $m = 0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ . Thus, we can define, for  $m =$   
 1133  $0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ ,

$$1134 \quad a_{m,\ell} := \theta_i, \quad \text{where } i = mL + \ell.$$

1135 Then, for  $m = 0, 1, \dots, M - 1$ , we set  $b_{m,0} = 0$  and  $b_{m,\ell} = a_{m,\ell-1}$  for  $\ell = 1, \dots, L - 1$ .

1136 By Lemma 5.6, there exist  $\phi_1, \phi_2 \in \mathcal{NN}(\text{width} \leq 4N + 3; \text{depth} \leq 3L + 3)$  such that

$$1137 \quad \phi_1(m, \ell) = \sum_{j=0}^{\ell} a_{m,j} \quad \text{and} \quad \phi_2(m, \ell) = \sum_{j=0}^{\ell} b_{m,j},$$

1138 for  $m = 0, 1, \dots, M - 1$  and  $\ell = 0, 1, \dots, L - 1$ .

1139 We consider the sample set

$$1140 \quad \{(mL, m) : m = 0, 1, \dots, M\} \cup \{((m+1)L - 1, m) : m = 0, 1, \dots, M - 1\}.$$

1141 Its size is  $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$ . By Lemma 5.4 (set  $N_1 = N$  and  $N_2 = 2NL - 1$   
 1142 therein), there exists

$$1143 \quad \begin{aligned} \psi &\in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL - 1) + 1]) \\ &= \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \end{aligned}$$

1144 such that

- 1145 •  $\psi(ML) = M$  and  $\psi(mL) = \psi((m+1)L - 1) = m$  for  $m = 0, 1, \dots, M - 1$ ;
- 1146 •  $\psi$  is linear on each interval  $[mL, (m+1)L - 1]$  for  $m = 0, 1, \dots, M - 1$ .

1147 It follows that

$$1148 \quad \psi(x) = m, \quad \text{if } x \in [mL, (m+1)L - 1], \quad \text{for } m = 0, 1, \dots, M - 1,$$

1149 implying

$$1150 \quad \psi(mL + \ell) = m \quad \text{for } m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1.$$

1151 For  $i = 0, 1, \dots, N^2L - 1$ , by representing  $i = mL + \ell$  for  $m = 0, 1, \dots, M - 1$  and  
 1152  $\ell = 0, 1, \dots, L - 1$ , we have  $\psi(i) = \psi(mL + \ell) = m$  and  $i - L\psi(i) = \ell$ , deducing

$$\begin{aligned} &\phi_1(\psi(i), i - L\psi(i)) - \phi_2(\psi(i), i - L\psi(i)) \\ &= \phi_1(m, \ell) - \phi_2(m, \ell) = \sum_{j=0}^{\ell} a_{m,j} - \sum_{j=0}^{\ell} b_{m,j} \\ 1153 \quad &= \sum_{j=0}^{\ell} a_{m,j} - \sum_{j=1}^{\ell} a_{m,j-1} - b_0 = a_{m,\ell} = \theta_i. \end{aligned} \tag{5.8}$$

1154 Therefore, the desired function  $\phi$  can be implemented by the network architecture  
 1155 described in Figure 14.

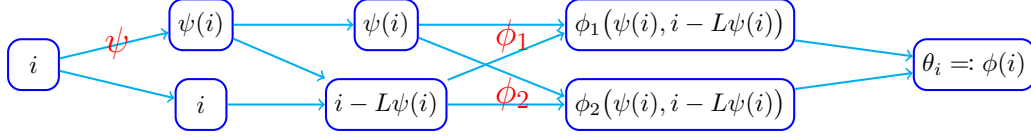


Figure 14: An illustration of the network architecture implementing the desired function  $\phi$  based on Equation (5.8).

Note that

$$\phi_1, \phi_2 \in \mathcal{NN}(\text{width} \leq 4N + 3; \text{depth} \leq 3L + 3).$$

And by Lemma 5.5,

$$\begin{aligned} \psi &\in \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \\ &\subseteq \mathcal{NN}(\text{width} \leq 4N + 2; \text{depth} \leq 2L + 1). \end{aligned}$$

Hence, the network architecture shown in Figure 14 is with width  $\max\{4L + 2 + 1, 2(4L + 3)\} = 8N + 6$  and depth  $(2L + 1) + 2 + (3L + 3) + 1 = 5L + 7$ , implying  $\phi \in \mathcal{NN}(\text{width} \leq 8N + 6; \text{depth} \leq 5L + 7)$ . So we finish the proof.  $\square$

With Lemma 5.7 in hand, we are now ready to prove Proposition 4.4.

*Proof of Proposition 4.4.* Set  $J = \lceil 2s \log_2(NL + 1) \rceil \in \mathbb{N}^+$ . For each  $\xi_i \in [0, 1]$ , there exist  $\xi_{i,1}, \xi_{i,2}, \dots, \xi_{i,J} \in \{0, 1\}$  such that

$$|\xi_i - \text{bin} 0.\xi_{i,1}\xi_{i,2}\dots\xi_{i,J}| \leq 2^{-J}, \quad \text{for } i = 0, 1, \dots, N^2L^2 - 1.$$

By Lemma 5.7, there exist

$$\phi_1, \phi_2, \dots, \phi_J \in \mathcal{NN}(\text{width} \leq 8N + 6; \text{depth} \leq 5L + 7)$$

such that

$$\phi_j(i) = \xi_{i,j}, \quad \text{for } i = 0, 1, \dots, N^2L^2 - 1 \text{ and } j = 1, 2, \dots, J.$$

Define

$$\tilde{\phi}(x) := \sum_{j=1}^J 2^{-j} \phi_j(x), \quad \text{for any } x \in \mathbb{R}.$$

It follows that, for  $i = 0, 1, \dots, N^2L^2 - 1$ ,

$$\begin{aligned} |\tilde{\phi}(i) - \xi_i| &= \left| \sum_{j=1}^J 2^{-j} \phi_j(i) - \xi_i \right| = \left| \sum_{j=1}^J 2^{-j} \xi_{i,j} - \xi_i \right| \\ &= |\text{bin} 0.\xi_{i,1}\xi_{i,2}\dots\xi_{i,J} - \xi_i| \leq 2^{-J} \leq N^{-2s} L^{-2s}, \end{aligned}$$

where the last inequality comes from

$$2^{-J} = 2^{-\lceil 2s \log_2(NL + 1) \rceil} \leq 2^{-2s \log_2(NL + 1)} = (NL + 1)^{-2s} \leq N^{-2s} L^{-2s}.$$

Now let us estimate the width and depth of the network implementing  $\tilde{\phi}$ . Recall that

$$\begin{aligned} J = \lceil 2s \log_2(NL + 1) \rceil &\leq 2s(1 + \log_2(NL + 1)) \leq 2s(1 + \log_2(2N) + \log_2 L) \\ &\leq 2s(1 + \log_2(2N))(1 + \log_2 L) \leq 2s \lceil \log_2(4N) \rceil \lceil \log_2(2L) \rceil, \end{aligned}$$

1180 and  $\phi_j \in \mathcal{NN}(\text{width} \leq 8N + 6; \text{depth} \leq 5L + 7)$  for each  $j$ . As shown in Figure 15,  
 1181  $\tilde{\phi} = \sum_{j=1}^J 2^{-j} \phi_j$  can be implemented by a ReLU FNN with width

1182 
$$(8N + 6)2s\lceil \log_2(4N) \rceil + 2s\lceil \log_2(4N) \rceil + 2 \leq 16s(N + 1)\log_2(8N)$$

1183 and depth

1184 
$$(5L + 7 + 1)\lceil \log_2(2L) \rceil \leq (5N + 8)\log_2(4L).$$

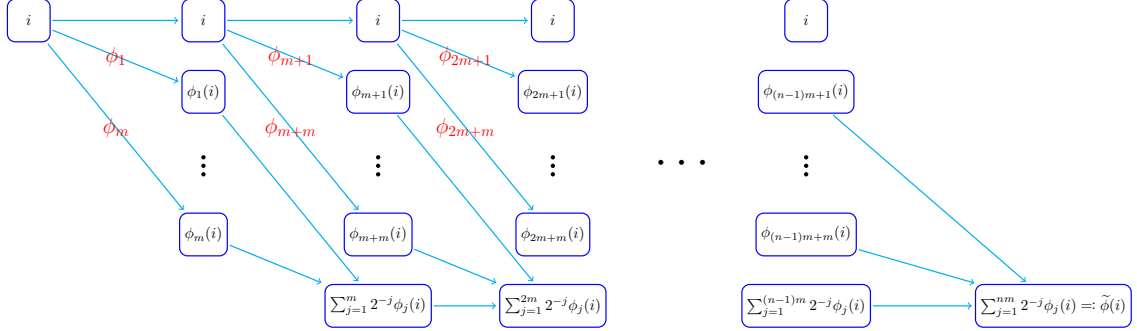


Figure 15: An illustration of the network architecture implementing  $\tilde{\phi} = \sum_{j=1}^J 2^{-j} \phi_j$ . We assume  $J = mn$ , where  $m = 2s\lceil \log_2(4N) \rceil$  and  $n = \lceil \log_2(2L) \rceil$ , since we can set  $\phi_{J+1} = \dots = \phi_{nm} = 0$  if  $J < nm$ . This network architecture can be interpreted as a ReLU one via simple modifications based on the fact  $x = \sigma(x) - \sigma(-x)$ .

1185 Finally, we define

1186 
$$\phi(x) := \min\{\sigma(\tilde{\phi}(x)), 1\} = \min\{\max\{0, \tilde{\phi}(x)\}, 1\}, \quad \text{for any } x \in \mathbb{R}.$$

1187 Then  $0 \leq \phi(x) \leq 1$  for any  $x \in \mathbb{R}$  and  $\phi$  can be implemented by a ReLU FNN with width  
 1188  $16s(N + 1)\log_2(8N)$  and depth  $(5L + 8)\log_2(4L) + 3 \leq 5(L + 2)\log_2(4L)$ . See Figure 16  
 1189 for the network architecture implementing  $\phi$ . Note that

1190 
$$\tilde{\phi}(i) = \sum_{j=1}^J 2^{-j} \phi_j(i) = \sum_{j=1}^J 2^{-j} \xi_{i,j} \in [0, 1], \quad \text{for } i = 0, 1, \dots, N^2 L^2 - 1.$$

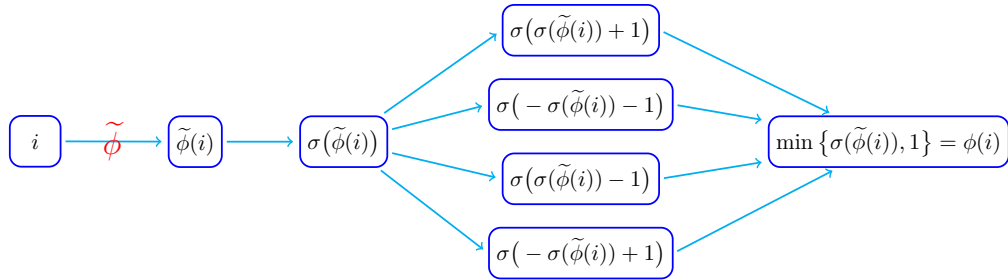


Figure 16: An illustration of the network architecture implementing the desired function  $\phi$  based on the fact  $\min\{x_1, x_2\} = \frac{x_1 + x_2 - |x_1 - x_2|}{2} = \frac{\sigma(x_1 + x_2) - \sigma(-x_1 - x_2) - \sigma(x_1 - x_2) - \sigma(-x_1 + x_2)}{2}$ .

1191

1192 It follows that

1193 
$$|\phi(i) - \xi_i| = \left| \min\{\max\{0, \tilde{\phi}(i)\}, 1\} - \xi_i \right| = |\tilde{\phi}(i) - \xi_i| \leq N^{-2s} L^{-2s},$$

1194 for  $i = 0, 1, \dots, N^2 L^2 - 1$ . The proof is complete.  $\square$

## 6 Conclusions

This paper has established a nearly optimal approximation rate of ReLU FNNs in terms of both width and depth to approximate smooth functions. It is shown that ReLU FNNs with width  $\mathcal{O}(N \ln N)$  and depth  $\mathcal{O}(L \ln L)$  can approximate functions in the unit ball of  $C^s([0, 1]^d)$  with approximation rate  $\mathcal{O}(N^{-2s/d} L^{-2s/d})$ . Through VC dimension, it is also proved that this approximation rate is asymptotically nearly tight for the closed unit ball of smooth function class  $C^s([0, 1]^d)$ .

We would like to remark that our analysis is for the fully connected feed-forward neural networks with the ReLU activation function. It would be an interesting direction to generalize our results to neural networks with other architectures (e.g., convolutional neural networks and ResNet) and activation functions (e.g., tanh and sigmoid functions). These will be left as future work.

## Acknowledgments

The work of J. Lu is supported in part by the National Science Foundation via grants DMS-1415939, CCF-1934964, and DMS-2012286. Z. Shen is supported by Tan Chin Tuan Centennial Professorship. H. Yang H. Yang was partially supported by the National Science Foundation under award DMS-1945029.

## References

- [1] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *ArXiv*, abs/1811.04918, 2019.
- [2] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
- [3] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *ICML*, 2019.
- [4] Chenglong Bao, Qianxiao Li, Zuowei Shen, Cheng Tai, Lei Wu, and Xueshuang Xiang. Approximation analysis of convolutional neural networks. 2019.
- [5] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.
- [6] Andrew R. Barron and Jason M. Klusowski. Approximation and estimation for high-dimensional deep learning networks, 2018.
- [7] Peter Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear VC dimension bounds for piecewise polynomial networks. *Neural Computation*, 10:217–3, 1998.



- 1229 [8] M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: A  
1230 comparison between shallow and deep architectures. *IEEE Transactions on Neural*  
1231 *Networks and Learning Systems*, 25(8):1553–1565, Aug 2014.
- 1232 [9] Helmut. Bölcskei, Philipp. Grohs, Gitta. Kutyniok, and Philipp. Petersen. Optimal  
1233 approximation with sparsely connected deep neural networks. *SIAM Journal on*  
1234 *Mathematics of Data Science*, 1(1):8–45, 2019.
- 1235 [10] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent  
1236 for wide and deep neural networks. *CoRR*, abs/1905.13210, 2019.
- 1237 [11] Liang Chen and Congwei Wu. A note on the expressive power of deep rectified linear  
1238 unit networks in high-dimensional spaces. *Mathematical Methods in the Applied*  
1239 *Sciences*, 42(9):3400–3404, 2019.
- 1240 [12] Minshuo Chen, Haoming Jiang, Wenjing Liao, and Tuo Zhao. Efficient approx-  
1241 imation of deep ReLU networks for functions on low dimensional manifolds. In  
1242 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett,  
1243 editors, *Advances in Neural Information Processing Systems 32*, pages 8174–8184.  
1244 Curran Associates, Inc., 2019.
- 1245 [13] Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much  
1246 over-parameterization is sufficient to learn deep ReLU networks? *CoRR*,  
1247 arXiv:1911.12360, 2019.
- 1248 [14] Charles K. Chui, Shao-Bo Lin, and Ding-Xuan Zhou. Construction of neural net-  
1249 works for realization of localized deep learning. *Frontiers in Applied Mathematics*  
1250 *and Statistics*, 4:14, 2018.
- 1251 [15] George Cybenko. Approximation by superpositions of a sigmoidal function. *MCSS*,  
1252 2:303–314, 1989.
- 1253 [16] Ronald A. Devore. Optimal nonlinear approximation. *Manuskripta Math*, pages  
1254 469–478, 1989.
- 1255 [17] Weinan E, Chao Ma, and Qingcan Wang. A priori estimates of the population risk  
1256 for residual networks. *ArXiv*, abs/1903.02154, 2019.
- 1257 [18] Weinan E, Chao Ma, and Lei Wu. A priori estimates of the population risk for  
1258 two-layer neural networks. *Communications in Mathematical Sciences*, 17(5):1407  
1259 – 1425, 2019.
- 1260 [19] Weinan E and Qingcan Wang. Exponential convergence of the deep neural network  
1261 approximation for analytic functions. *CoRR*, abs/1807.00297, 2018.
- 1262 [20] Rémi Gribonval, Gitta Kutyniok, Morten Nielsen, and Felix Voigtlaender. Approx-  
1263 imation spaces of deep neural networks. *arXiv e-prints*, page arXiv:1905.01208, May  
1264 2019.

- [21] Ingo Gühring, Gitta Kutyniok, and Philipp Petersen. Error bounds for approximations with deep ReLU neural networks in  $W^{s,p}$  norms. *arXiv e-prints*, page arXiv:1902.07896, Feb 2019.
- [22] Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1064–1068, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.
- [23] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.
- [24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.
- [25] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018.
- [26] Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks. *ArXiv*, abs/1909.12292, 2020.
- [27] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. *J. Comput. Syst. Sci.*, 48(3):464–497, June 1994.
- [28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.
- [29] Qianxiao Li, Ting Lin, and Zuowei Shen. Deep Learning via Dynamical Systems: An Approximation Perspective. *arXiv e-prints*, page arXiv:1912.10382, December 2019.
- [30] Shiyu Liang and R. Srikant. Why deep neural networks? *CoRR*, abs/1610.04161, 2016.
- [31] Hadrien Montanelli and Qiang Du. New error bounds for deep networks using sparse grids. 2017.
- [32] Hadrien Montanelli and Haizhao Yang. Error bounds for deep ReLU networks using the kolmogorov–arnold superposition theorem. *Neural Networks*, 129:1 – 6, 2020.
- [33] Hadrien Montanelli, Haizhao Yang, and Qiang Du. Deep ReLU networks overcome the curse of dimensionality for bandlimited functions. 2019.
- [34] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2924–2932. Curran Associates, Inc., 2014.

- 1303 [35] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and estimation of  
1304 deep neural network with intrinsic dimensionality. 2019.
- 1305 [36] J. A. A. Opschoor, Ch. Schwab, and J. Zech. Exponential ReLU DNN expression  
1306 of holomorphic maps in high dimension. Technical Report 2019-35, Seminar for  
1307 Applied Mathematics, ETH Zürich, Switzerland., 2019.
- 1308 [37] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise  
1309 smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296  
1310 – 330, 2018.
- 1311 [38] T. Poggio, H. N. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when  
1312 can deep—but not shallow—networks avoid the curse of dimensionality: A review.  
1313 *International Journal of Automation and Computing*, 14:503–519, 2017.
- 1314 [39] Akito Sakurai. Tight bounds for the VC-dimension of piecewise polynomial net-  
1315 works. In *Advances in Neural Information Processing Systems*, pages 323–329.  
1316 Neural information processing systems foundation, 1999.
- 1317 [40] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Nonlinear approximation via com-  
1318 positions. *Neural Networks*, 119:74 – 84, 2019.
- 1319 [41] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation  
1320 characterized by number of neurons. *Communications in Computational Physics*,  
1321 28(5):1768–1811, 2020.
- 1322 [42] Taiji Suzuki. Adaptivity of deep reLU network for learning in Besov and mixed  
1323 smooth Besov spaces: optimal rate and curse of dimensionality. In *International  
1324 Conference on Learning Representations*, 2019.
- 1325 [43] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks.  
1326 *Neural Networks*, 94:103 – 114, 2017.
- 1327 [44] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep  
1328 ReLU networks. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet,  
1329 editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Pro-  
1330 ceedings of Machine Learning Research*, pages 639–649. PMLR, 06–09 Jul 2018.
- 1331 [45] Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation  
1332 rates for deep neural networks. *arXiv e-prints*, page arXiv:1906.09477, Jun 2019.
- 1333 [46] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and  
1334 Computational Harmonic Analysis*, 2019.