

# Optimal Approximation Rate of ReLU Networks in terms of Width and Depth\*

Zuowei Shen<sup>†</sup>      Haizhao Yang<sup>‡</sup>      Shijun Zhang<sup>§</sup>

## Abstract

This paper concentrates on the approximation power of deep feed-forward neural networks in terms of width and depth. It is proved by construction that ReLU networks with width  $\mathcal{O}(\max\{d\lfloor N^{1/d} \rfloor, N+2\})$  and depth  $\mathcal{O}(L)$  can approximate a Hölder continuous function on  $[0, 1]^d$  with an approximation rate  $\mathcal{O}(\lambda\sqrt{d}(N^2L^2\ln N)^{-\alpha/d})$ , where  $\alpha \in (0, 1]$  and  $\lambda > 0$  are Hölder order and constant, respectively. Such a rate is optimal up to a constant in terms of width and depth separately, while existing results are only nearly optimal without the logarithmic factor in the approximation rate. More generally, for an arbitrary continuous function  $f$  on  $[0, 1]^d$ , the approximation rate becomes  $\mathcal{O}(\sqrt{d}\omega_f((N^2L^2\ln N)^{-1/d}))$ , where  $\omega_f(\cdot)$  is the modulus of continuity. We also extend our analysis to any continuous function  $f$  on a bounded set. Particularly, if ReLU networks with depth 31 and width  $\mathcal{O}(N)$  are used to approximate one-dimensional Lipschitz continuous functions on  $[0, 1]$  with a Lipschitz constant  $\lambda > 0$ , the approximation rate in terms of the total number of parameters,  $W = \mathcal{O}(N^2)$ , becomes  $\mathcal{O}(\frac{\lambda}{W\ln W})$ , which has not been discovered in the literature for fixed-depth ReLU networks.

**Key words.** Deep ReLU Networks; Optimal Approximation; VC-dimension; Bit Extraction.

## 1 Introduction

Over the past few decades, the expressiveness of neural networks has been widely studied from many points of view, e.g., in terms of combinatorics [27], topology [4], Vapnik-Chervonenkis (VC) dimension [3, 13, 31], fat-shattering dimension [1, 19], information theory [30], classical approximation theory [2, 6, 11, 16, 20, 24, 32, 32–36, 41, 44], optimization [14, 17, 18, 21, 29]. The error analysis of neural networks consists of three parts: the approximation error, the optimization error, and the generalization error. This paper focuses on the approximation error for ReLU networks.

The approximation errors of feed-forward neural networks with various activation functions have been studied for different types of functions, e.g., smooth functions

---

\*Submitted to the editors DATE.

<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)).

[9, 22, 24, 25, 40], piecewise smooth functions [30], band-limited functions [26], continuous functions [33–35, 41]. In the early works of approximation theory for neural networks, the universal approximation theorem [6, 15, 16] without approximation rates showed that there exists a sufficiently large neural network approximating a target function in a certain function space within any given error  $\varepsilon > 0$ . In particular, it is shown in [23] that the ReLU-activated residual neural network with one-neuron hidden layers is a universal approximator. The universal approximation property for general residual neural networks was proved in [20] via a dynamical system approach.

An asymptotic analysis of the approximation rate in terms of depth is provided in [41, 43] for ReLU networks. To be exact, the nearly optimal approximation rates of ReLU networks with width  $\mathcal{O}(d)$  and depth  $\mathcal{O}(L)$  for functions in  $C([0, 1]^d)$  and  $C^s([0, 1]^d)$  are  $\mathcal{O}(\omega_f(L^{-2/d}))$  and  $\mathcal{O}((L/\ln L)^{-2s/d})$ , respectively. These two papers provide the approximation rate in terms of depth asymptotically for fixed-width networks. A different approach is used in [24, 33] to obtain a quantitative characterization of the approximation rate in terms of width, depth, and smoothness order for continuous and smooth functions.

Particularly, it was shown in [33] that a ReLU network with width  $C_1(d) \cdot N$  and depth  $C_2(d) \cdot L$  can attain an approximation error  $C_3(d) \cdot \omega_f(N^{-2/d} L^{-2/d})$  to approximate a continuous function  $f$  on  $[0, 1]^d$ , where  $C_1(d)$ ,  $C_2(d)$ , and  $C_3(d)$  are three constants in  $d$  with explicit formulas to specify their values, and  $\omega_f(\cdot)$  is the modulus of continuity of  $f \in C([0, 1]^d)$  defined via

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

Such an approximation rate is optimal in terms of  $N$  and  $L$  up to a logarithmic term and the corresponding optimal approximation theory is still unavailable. To address this problem, we provide a constructive proof in this paper to show that ReLU networks of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  can approximate an arbitrary continuous function  $f$  on  $[0, 1]^d$  with an optimal approximation error  $\mathcal{O}(\sqrt{d} \omega_f((N^2 L^2 \ln N)^{-\alpha/d}))$  in terms of  $N$  and  $L$ . As shown by our main result, Theorem 1.1 below, the approximation rate obtained here admits explicit formulas to specify its prefactors when  $\omega_f(\cdot)$  is known.

**Theorem 1.1.** *Given a continuous function  $f \in C([0, 1]^d)$ , for any  $N \in \mathbb{N}^+$ ,  $L \in \mathbb{N}^+$ , and  $p \in [1, \infty]$ , there exists a function  $\phi$  implemented by a ReLU network with width  $C_1 \max\{d \lfloor N^{1/d} \rfloor, N + 2\}$  and depth  $11L + C_2$  such that*

$$\|f - \phi\|_{L^p([0, 1]^d)} \leq 131 \sqrt{d} \omega_f \left( (N^2 L^2 \log_3(N + 2))^{-1/d} \right),$$

where  $C_1 = 16$  and  $C_2 = 18$  if  $p \in [1, \infty)$ ;  $C_1 = 3^{d+3}$  and  $C_2 = 18 + 2d$  if  $p = \infty$ .

Note that  $3^{d+3} \max\{d \lfloor N^{1/d} \rfloor, N + 2\} \leq 3^{d+3} \max\{dN, 3N\} \leq 3^{d+4} dN$ . Given any  $\tilde{N}, \tilde{L} \in \mathbb{N}^+$  with  $\tilde{N} \geq 3^{d+4} d$  and  $\tilde{L} \geq 29 + 2d$ , there exist  $N, L \in \mathbb{N}^+$  such that

$$3^{d+4} dN \leq \tilde{N} < 3^{d+4} d(N + 1) \quad \text{and} \quad 11L + 18 + 2d \leq \tilde{L} < 11(L + 1) + 18 + 2d.$$

If follows that

$$N \geq \frac{N + 1}{3} > \frac{\tilde{N}}{3^{d+5} d} \quad \text{and} \quad L \geq \frac{L + 1}{2} > \frac{1}{2} \cdot \frac{\tilde{L} - 18 - 2d}{11} = \frac{\tilde{L} - 18 - 2d}{22}.$$

Then we have an immediate corollary of Theorem 1.1.

**Corollary 1.2.** *Given a continuous function  $f \in C([0, 1]^d)$ , for any  $\tilde{N} \in \mathbb{N}^+$  and  $\tilde{L} \in \mathbb{N}^+$  with  $\tilde{N} \geq 3^{d+4}d$  and  $\tilde{L} \geq 29+2d$ , there exists a function  $\phi$  implemented by a ReLU network with width  $\tilde{N}$  and depth  $\tilde{L}$  such that*

$$\|f - \phi\|_{L^\infty([0,1]^d)} \leq 131\sqrt{d}\omega_f\left(\left(\left(\frac{\tilde{N}}{3^{d+5}d}\right)^2\left(\frac{\tilde{L}-18-2d}{22}\right)^2\log_3\left(\frac{\tilde{N}}{3^{d+5}d} + 2\right)\right)^{-1/d}\right).$$

As a special case of Theorem 1.1 for explicit error characterization, let us take Hölder continuous functions as an example. Let  $\text{Hölder}([0, 1]^d, \alpha, \lambda)$  denote the space of Hölder continuous functions on  $[0, 1]^d$  of order  $\alpha \in (0, 1]$  with a Hölder constant  $\lambda > 0$ . We have an immediate corollary of Theorem 1.1 as follows.

**Corollary 1.3.** *Given a Hölder continuous function  $f \in \text{Hölder}([0, 1]^d, \alpha, \lambda)$ , for any  $N \in \mathbb{N}^+$ ,  $L \in \mathbb{N}^+$ , and  $p \in [1, \infty]$ , there exists a function  $\phi$  implemented by a ReLU network with width  $C_1 \max\{d\lfloor N^{1/d} \rfloor, N+2\}$  and depth  $11L + C_2$  such that*

$$\|f - \phi\|_{L^p([0,1]^d)} \leq 131\lambda\sqrt{d}\left(N^2L^2\log_3(N+2)\right)^{-\alpha/d},$$

where  $C_1 = 16$  and  $C_2 = 18$  if  $p \in [1, \infty)$ ;  $C_1 = 3^{d+3}$  and  $C_2 = 18 + 2d$  if  $p = \infty$ .

To better illustrate the importance of our theory, we summarize our key contributions as follows.

(1) Upper bound: We provide a quantitative and non-asymptotic approximation rate  $131\sqrt{d}\omega_f\left(\left(N^2L^2\log_3(N+2)\right)^{-1/d}\right)$  in terms of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  for any  $f \in C([0, 1]^d)$  in Theorem 1.1.

(1.1) This approximation error analysis can be extended to  $f \in C(E)$  for any  $E \subseteq [-R, R]^d$  with  $R > 0$  as we shall see later in Theorem 2.5.

(1.2) In the case of one-dimensional Lipschitz continuous functions on  $[0, 1]$  with a Lipschitz constant  $\lambda > 0$ , the approximation rate in Theorem 1.1 becomes  $\mathcal{O}\left(\frac{\lambda}{W \ln W}\right)$  for ReLU networks with 31 hidden layers and  $\mathcal{O}(W)$  parameters via setting  $L = 1$  and  $W = \mathcal{O}(N^2)$  therein. To the best of our knowledge, the approximation rate  $\mathcal{O}\left(\frac{\lambda}{W \ln W}\right)$  is better than existing known results using fixed-depth ReLU networks to approximate Lipschitz continuous functions on  $[0, 1]$ .

(2) Lower bound: Through the VC-dimension bounds of ReLU networks given in [13], we show, in Section 2.3, that the approximation rate  $131\lambda\sqrt{d}\left(N^2L^2\log_3(N+2)\right)^{-\alpha/d}$  in terms of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  for  $\text{Hölder}([0, 1]^d, \alpha, \lambda)$  is optimal as follows.

(2.1) When the width is fixed, both the approximation upper and lower bounds take the form of  $CL^{-2\alpha/d}$  for a positive constant  $C$ .

(2.2) When the depth is fixed, both the approximation upper and lower bounds take the form of  $C(N^2 \ln N)^{-\alpha/d}$  for a positive constant  $C$ .

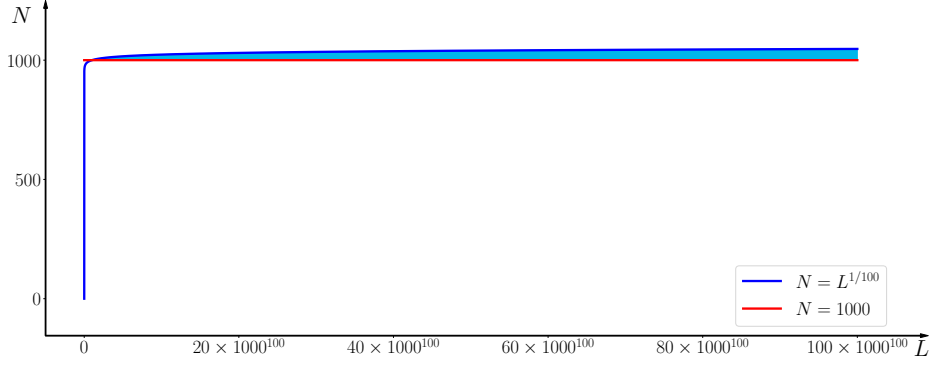


Figure 1: Our rate is optimal in terms of width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  simultaneously except for the region marked in cyan characterized by  $\{(N, L) \in \mathbb{N}^2 : C_1 \leq N \leq L^{C_2}\}$ , where  $C_i = C_i(\alpha, d)$  for  $i = 1, 2$  are two positive constants. This figure is an example for  $C_1 = 1000$  and  $C_2 = 1/100$ .

We would like to point out that if  $N$  and  $L$  vary simultaneously, the rate is optimal in the  $N$ - $L$  plane except for a small region as shown in Figure 1. See Section 2.3 for a detailed discussion. The earlier result in [33] provides a nearly optimal approximation error that has a gap (a logarithmic term) between the lower and upper bounds. It is technically challenging to match the upper bound with the lower bound. Compared to the nearly optimal rate  $19\lambda\sqrt{d}N^{-2\alpha/d}L^{-2\alpha/d}$  for Hölder continuous functions in Hölder( $[0, 1]^d, \alpha, \lambda$ ) in [33], this paper achieves the optimal rate  $131\lambda\sqrt{d}(N^2L^2\log_3(N+2))^{-\alpha/d}$  using more technical and sophisticated construction. For example, a novel bit extraction technique different to that in [3] is proposed, and new ReLU networks are constructed to approximate step functions more efficiently than those in [33]. The optimal result obtained in this paper could also be extended to other functions spaces, leading to better understanding of deep network approximation.

We have obtained the optimal approximation rate for (Hölder) continuous functions approximated by ReLU networks. There are two possible directions to improve the approximation rate or reduce the effect of the curse of dimensionality. The first one is to consider proper target function spaces, e.g., Barron spaces [2, 8, 12, 37], band-limited functions [5, 26], smooth functions [24, 43], and analytic functions [9]. The other direction is to consider neural networks with other activation functions. For example, the results of [43] imply that (sin, ReLU)-activated networks with  $W$  parameters can achieve an asymptotic approximation error  $\mathcal{O}(2^{-c_d\sqrt{W}})$  for Lipschitz continuous functions defined on  $[0, 1]^d$ , where  $c_d$  is an unknown constant depending on  $d$ . Floor-ReLU networks with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  are constructed in [34] to admit an approximation rate  $\mathcal{O}(\omega_f(\sqrt{d}N^{-\sqrt{L}}))$  for any continuous function  $f \in C([0, 1]^d)$ . It is shown in [35] that three-hidden-layer networks with  $\mathcal{O}(W)$  parameters using the floor function ( $\lfloor x \rfloor$ ), the exponential function ( $2^x$ ), and the step function ( $\mathbb{1}_{x \geq 0}$ ) as activation functions can approximate Lipschitz functions defined on  $[0, 1]^d$  with an exponentially small error  $\mathcal{O}(\sqrt{d}2^{-W})$ . By the use of more sophisticated activation functions instead of those used in [34, 35, 43], a recent paper [42] shows that there exists a network of size depending on  $d$  implicitly, achieving an arbitrary approximation error for any continuous function in  $C([0, 1]^d)$ . A key ingredient of the approaches mentioned above is to use more than one

136 activation functions to design neural network architectures.

137 The error analysis of deep learning is to estimate approximation, generalization, and  
 138 optimization errors. Here, we give a brief discussion, the interested reader can find more  
 139 details in [24, 34]. Let  $\phi(\mathbf{x}; \boldsymbol{\theta})$  denote a function computed by a network parameterized  
 140 with  $\boldsymbol{\theta}$ . Given a target function  $f$ , the final goal is to find the expected risk minimizer

$$141 \quad \boldsymbol{\theta}_{\mathcal{D}} := \arg \min_{\boldsymbol{\theta}} R_{\mathcal{D}}(\boldsymbol{\theta}), \quad \text{where } R_{\mathcal{D}}(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x} \sim U(\mathcal{X})} [\ell(\phi(\mathbf{x}; \boldsymbol{\theta}), f(\mathbf{x}))],$$

142 with a loss function  $\ell(\cdot, \cdot)$  and an unknown data distribution  $U(\mathcal{X})$ .

143 In practice, for given samples  $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^n$ , the goal of supervised learning is to  
 144 identify the empirical risk minimizer

$$145 \quad \boldsymbol{\theta}_{\mathcal{S}} := \arg \min_{\boldsymbol{\theta}} R_{\mathcal{S}}(\boldsymbol{\theta}), \quad \text{where } R_{\mathcal{S}}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n \ell(\phi(\mathbf{x}_i; \boldsymbol{\theta}), f(\mathbf{x}_i)).$$

146 In fact, one could only get a numerical minimizer  $\boldsymbol{\theta}_{\mathcal{N}}$  via a numerical optimization  
 147 method. The discrepancy between the target function  $f$  and the learned function  
 148  $\phi(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{N}})$  is measured by  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$ , which is bounded by

$$149 \quad R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) \leq \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{Approximation error}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{Optimization error}} + \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})] + [R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{Generalization error}}.$$

150 This paper deals with the approximation error of ReLU networks for continuous functions  
 151 and gives an upper bound of  $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$  which is optimal up to a constant. Note that  
 152 the approximation error analysis given here is independent of data samples and deep  
 153 learning algorithms. However, the analysis of optimization and generalization errors  
 154 do depend on data samples, deep learning algorithms, models, etc. For example, refer  
 155 to [7, 8, 10, 14, 17, 18, 21, 28, 29] for a further understanding of the generalization and  
 156 optimization errors.

157 The rest of this paper is organized as follows. In Section 2, we prove Theorem 1.1  
 158 by assuming Theorem 2.1 is true, show the optimality of Theorem 1.1, and extend our  
 159 analysis to continuous functions defined on any bounded set. Next, Theorem 2.1 is  
 160 proved in Section 3 based on Proposition 3.1 and 3.2, the proofs of which can be found  
 161 in Section 4. Finally, Section 5 concludes this paper with a short discussion.

## 162 2 Theoretical analysis

163 In this section, we first prove Theorem 1.1 and discuss its optimality. Next, we ex-  
 164 tend our analysis to general continuous functions defined on any bounded set. Notations  
 165 throughout this paper are summarized in Section 2.1.

### 166 2.1 Notations

167 Let us summarize all basic notations used in this paper as follows.

- 168 • Matrices are denoted by bold uppercase letters. For instance,  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is a real  
 169 matrix of size  $m \times n$ , and  $\mathbf{A}^T$  denotes the transpose of  $\mathbf{A}$ . Vectors are denoted

as bold lowercase letters. For example,  $\mathbf{v} = [v_1, \dots, v_d]^T = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \in \mathbb{R}^d$  is a column vector with  $\mathbf{v}(i) = v_i$  being the  $i$ -th element. Besides, “[” and “]” are used to partition matrices (vectors) into blocks, e.g.,  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$ .

- For any  $p \in [1, \infty)$ , the  $p$ -norm (or  $\ell^p$ -norm) of a vector  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d$  is defined by

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

- 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  $+\infty$ .

- 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.,  $\text{bin}0.\theta_1\theta_2\cdots\theta_L := \sum_{\ell=1}^L \theta_{\ell} 2^{-\ell}$ .

- Let  $\mu(\cdot)$  denote the Lebesgue measure.

- Let  $\mathbb{1}_S$  be the characteristic function on a set  $S$ , i.e.,  $\mathbb{1}_S$  is equal to 1 on  $S$  and 0 outside  $S$ .

- Let  $|S|$  denote the size of a set  $S$ , i.e., the number of all elements in  $S$ .

- The set difference of two sets  $A$  and  $B$  is denoted by  $A \setminus B := \{x : x \in A, x \notin B\}$ .

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

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

In particular,  $\Omega([0, 1]^d, K, \delta) = \emptyset$  if  $K = 1$ . See Figure 2 for two examples of trifling regions.

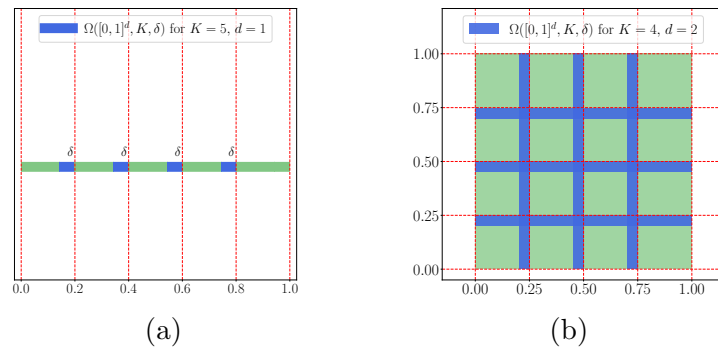


Figure 2: Two examples of trifling regions. (a)  $K = 5, d = 1$ . (b)  $K = 4, d = 2$ .

- Let  $\text{H\"older}([0, 1]^d, \alpha, \lambda)$  denote the space of H\"older continuous functions on  $[0, 1]^d$  of order  $\alpha \in (0, 1]$  with a H\"older constant  $\lambda > 0$ .

- 194 • For a continuous piecewise linear function  $f(x)$ , the  $x$  values where the slope  
195 changes are typically called **breakpoints**.
- 196 • Let  $\text{CPwL}(\mathbb{R}, n)$  denote the space that consists of all continuous piecewise linear  
197 functions with at most  $n$  breakpoints on  $\mathbb{R}$ .
- 198 • Let  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  denote the rectified linear unit (ReLU), i.e.  $\sigma(x) = \max\{0, x\}$ . With  
199 a slight abuse of notation, 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  
200  $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$ .
- 201 • We will use  $\mathcal{NN}$  to denote a function implemented by a ReLU network for short  
202 and use Python-type notations to specify a class of functions implemented by  
203 ReLU networks with several conditions, e.g.,  $\mathcal{NN}(c_1; c_2; \dots; c_m)$  is a set of func-  
204 tions implemented by ReLU networks satisfying  $m$  conditions given by  $\{c_i\}_{1 \leq i \leq m}$ ,  
205 each of which may specify the number of inputs (#input), the number of outputs  
206 (#output), the number of hidden layers (depth), the total number of parameters  
207 (#parameter), and the width in each hidden layer (widthvec), the maximum width  
208 of all hidden layers (width), etc. For example, if  $\phi \in \mathcal{NN}(\text{\#input} = 2; \text{\text{widthvec}} =$   
209  $[100, 100]; \text{\#output} = 1)$ , then  $\phi$  is a functions satisfies
  - 210 –  $\phi$  maps from  $\mathbb{R}^2$  to  $\mathbb{R}$ .
  - 211 –  $\phi$  can be implemented by a ReLU network with two hidden layers and the  
212 number of neurons in each hidden layer is 100.
- 213 • For any function  $\phi \in \mathcal{NN}(\text{\#input} = d; \text{\text{widthvec}} = [N_1, N_2, \dots, N_L]; \text{\#output} = 1)$ ,  
214 if we set  $N_0 = d$  and  $N_{L+1} = 1$ , then the architecture of the network implementing  
215  $\phi$  can be briefly described as follows:

$$216 \quad \mathbf{x} = \tilde{\mathbf{h}}_0 \xrightarrow[\mathcal{L}_0]{\mathbf{W}_0, \mathbf{b}_0} \mathbf{h}_1 \xrightarrow{\sigma} \tilde{\mathbf{h}}_1 \dots \xrightarrow[\mathcal{L}_{L-1}]{\mathbf{W}_{L-1}, \mathbf{b}_{L-1}} \mathbf{h}_L \xrightarrow{\sigma} \tilde{\mathbf{h}}_L \xrightarrow[\mathcal{L}_L]{\mathbf{W}_L, \mathbf{b}_L} \mathbf{h}_{L+1} = \phi(\mathbf{x}),$$

217 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  
218 the  $i$ -th affine linear transform  $\mathcal{L}_i$ , respectively, i.e.,

$$219 \quad \mathbf{h}_{i+1} = \mathbf{W}_i \cdot \tilde{\mathbf{h}}_i + \mathbf{b}_i =: \mathcal{L}_i(\tilde{\mathbf{h}}_i), \quad \text{for } i = 0, 1, \dots, L,$$

220 and

$$221 \quad \tilde{\mathbf{h}}_i = \sigma(\mathbf{h}_i), \quad \text{for } i = 1, 2, \dots, L.$$

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

$$223 \quad \phi = \mathcal{L}_L \circ \sigma \circ \mathcal{L}_{L-1} \circ \sigma \circ \dots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0,$$

224 which has been illustrated in Figure 3.

- 225 • The expression “a network with width  $N$  and depth  $L$ ” means
  - 226 – The maximum width of this network for all **hidden** layers is no more than  
227  $N$ .
  - 228 – The number of **hidden** layers of this network is no more than  $L$ .



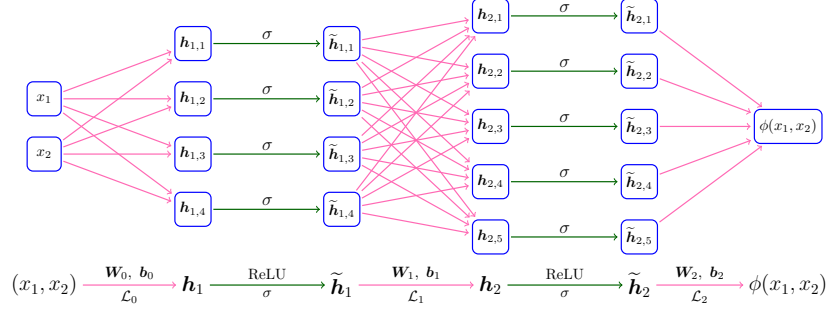


Figure 3: An example of a ReLU network with width 5 and depth 2.

## 2.2 Proof of Theorem 1.1

The key point is to construct piecewise constant functions to approximate continuous functions in the proof. However, it is impossible to construct a piecewise constant function implemented by a ReLU network due to the continuity of ReLU networks. Thus, we introduce the trifling region  $\Omega([0, 1]^d, K, \delta)$ , defined in Equation (2.1), and use ReLU networks to implement piecewise constant functions outside the trifling region. To prove Theorem 1.1, we first introduce a weaker variant of Theorem 1.1, showing how to construct ReLU networks to pointwisely approximate continuous functions except for the trifling region.

**Theorem 2.1.** *Given a function  $f \in C([0, 1]^d)$ , for any  $N \in \mathbb{N}^+$  and  $L \in \mathbb{N}^+$ , there exists a function  $\phi$  implemented by a ReLU network with width  $\max\{8d\lfloor N^{1/d} \rfloor + 3d, 16N + 30\}$  and depth  $11L + 18$  such that  $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$  and*

$$|f(\mathbf{x}) - \phi(\mathbf{x})| \leq 130\sqrt{d}\omega_f\left((N^2L^2\log_3(N+2))^{-1/d}\right), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

where  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor \log_3(N+2) \rfloor^{1/d}$  and  $\delta$  is an arbitrary number in  $(0, \frac{1}{3K}]$ .

With Theorem 2.1 that will be proved in Section 3, we can easily prove Theorem 1.1 for the case  $p \in [1, \infty)$ . To attain the rate in  $L^\infty$ -norm, we need to control the approximation error in the trifling region. To this end, we introduce a theorem to deal with the approximation inside the trifling region  $\Omega([0, 1]^d, K, \delta)$ .

**Theorem 2.2** (Theorem 3.7 of [44] or Theorem 2.1 of [24]). *Given any  $\varepsilon > 0$ ,  $N, L, K \in \mathbb{N}^+$ , and  $\delta \in (0, \frac{1}{3K}]$ , assume  $f$  is a continuous function in  $C([0, 1]^d)$  and  $\tilde{\phi}$  can be implemented by a ReLU network with width  $N$  and depth  $L$ . If*

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

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

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

Now we are ready to prove Theorem 1.1 by assuming Theorem 2.1 is true, which will be proved later in Section 3.



256 *Proof of Theorem 1.1.* We may assume  $f$  is not a constant function since it is a trivial  
 257 case. Then  $\omega_f(r) > 0$  for any  $r > 0$ . Let us first consider the case  $p \in [1, \infty)$ . Set  
 258  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor \log_3(N+2) \rfloor^{1/d}$  and choose a small  $\delta \in (0, \frac{1}{3K}]$  such that

$$\begin{aligned} Kd\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p &= \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor \log_3(N+2) \rfloor^{1/d} d\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p \\ &\leq \left( \omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right) \right)^p. \end{aligned}$$

260 By Theorem 2.1, there exists a function  $\phi$  implemented by a ReLU network with width

$$261 \quad \max\{8d\lfloor N^{1/d} \rfloor + 3d, 16N + 30\} \leq 16 \max\{d\lfloor N^{1/d} \rfloor, N + 2\}$$

262 and depth  $11L + 18$  such that  $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$  and

$$263 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq 130\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

264 It follows from  $\mu(\Omega([0, 1]^d, K, \delta)) \leq Kd\delta$  and  $\|f\|_{L^\infty([0, 1]^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$  that

$$\begin{aligned} \|f - \phi\|_{L^p([0, 1]^d)}^p &= \int_{\Omega([0, 1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} + \int_{[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} \\ &\leq Kd\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p + \left(130\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right)\right)^p \\ &\leq \left(\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right)\right)^p + \left(130\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right)\right)^p \\ &\leq \left(131\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right)\right)^p. \end{aligned}$$

266 Hence,  $\|f - \phi\|_{L^p([0, 1]^d)} \leq 131\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right)$ .

267 Next, let us discuss the case  $p = \infty$ . Set  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor \log_3(N+2) \rfloor^{1/d}$  and  
 268 choose a small  $\delta \in (0, \frac{1}{3K}]$  such that

$$269 \quad d \cdot \omega_f(\delta) \leq \omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right).$$

270 By Theorem 2.1, there exists a function  $\tilde{\phi}$  implemented by a ReLU network with width  
 271  $\max\{8d\lfloor N^{1/d} \rfloor + 3d, 16N + 30\}$  and depth  $11L + 18$  such that

$$272 \quad |f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq 130\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right) =: \varepsilon,$$

273 for any  $\mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ . By Theorem 2.2, there exists a function  $\phi$  imple-  
 274 mented by a ReLU network with width

$$275 \quad 3^d \left( \max\{8d\lfloor N^{1/d} \rfloor + 3d, 16N + 30\} + 4 \right) \leq 3^{d+3} \max\{d\lfloor N^{1/d} \rfloor, N + 2\}$$

276 and depth  $11L + 18 + 2d$  such that

$$277 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta) \leq 131\sqrt{d}\omega_f\left((N^2 L^2 \log_3(N+2))^{-1/d}\right), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

278 So we finish the proof.  $\square$

## 2.3 Optimality

This section will show that the approximation rates in Theorem 1.1 and Corollary 1.3 are optimal and there is no room to improve for the function class  $\text{Hölder}([0, 1]^d, \alpha, \lambda)$ . Therefore, the approximation rate for the whole continuous functions space in terms of width and depth in Theorem 1.1 cannot be improved. A typical method to characterize the optimal approximation theory of neural networks is to study the connection between the approximation error and Vapnik–Chervonenkis (VC) dimension [24, 33, 40, 41, 44]. This method relies on the VC-dimension upper bound given in [13]. In this paper, we adopt this method with several modifications to simplify the proof.

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

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

where  $|\cdot|$  denotes the size of a set. This equation means, given any  $\theta_i \in \{0, 1\}$  for  $i = 1, 2, \dots, m$ , there exists  $h \in H$  such that  $h(\mathbf{x}_i) = \theta_i$  for all  $i$ . For a general function set  $\mathcal{F}$  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

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

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

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

**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,

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

if  $\{m \in \mathbb{N}^+ : \Pi_H(m) = 2^m\}$  is not empty. In the case of  $\{m \in \mathbb{N}^+ : \Pi_H(m) = 2^m\} = \emptyset$ , we may define  $\text{VCDim}(H) = 0$ .

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).

We remark that one may also define  $\text{VCDim}(\mathcal{F})$  as  $\text{VCDim}(\mathcal{F}) := \text{VCDim}(\tilde{\mathcal{T}} \circ \mathcal{F})$ , where

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

313 Note that function spaces generated by networks are closed under linear transformation.  
 314 Thus, these two definitions of VC-dimension are equivalent.

315 The theorem below, similar to Theorem 4.17 of [44], reveals the connection between  
 316 VC-dimension and approximation rate.

317 **Theorem 2.4.** Assume  $\mathcal{F}$  is a set of functions mapping from  $[0, 1]^d$  to  $\mathbb{R}$ . For any  
 318  $\varepsilon > 0$ , if  $\text{VCDim}(\mathcal{F}) \geq 1$  and

$$319 \quad \inf_{\phi \in \mathcal{F}} \|\phi - f\|_{L^\infty([0,1]^d)} \leq \varepsilon, \quad \text{for any } f \in \text{Hölder}([0, 1]^d, \alpha, 1), \quad (2.2)$$

320 then  $\text{VCDim}(\mathcal{F}) \geq (9\varepsilon)^{-d/\alpha}$ .

321 This theorem demonstrates the connection between VC-dimension of  $\mathcal{F}$  and the ap-  
 322 proximation rate using elements of  $\mathcal{F}$  to approximate functions in  $\text{Hölder}([0, 1]^d, \alpha, \lambda)$ .  
 323 To be precise, the VC-dimension of  $\mathcal{F}$  determines an approximation rate lower bound  
 324  $\text{VCDim}(\mathcal{F})^{-\alpha/d}/9$ , which is the best possible approximation rate. Denote the best ap-  
 325 proximation error of functions in  $\text{Hölder}([0, 1]^d, \alpha, 1)$  approximated by ReLU networks  
 326 with width  $N$  and depth  $L$  as

$$327 \quad \mathcal{E}_{\alpha,d}(N, L) := \sup_{f \in \text{Hölder}([0,1]^d, \alpha, 1)} \left( \inf_{\phi \in \mathcal{NN}(\text{width} \leq N; \text{depth} \leq L)} \|\phi - f\|_{L^\infty([0,1]^d)} \right),$$

328 We have three remarks listed below.

329 (i) A large VC-dimension cannot guarantee a good approximation rate. For example,  
 330 it is easy to verify that

$$331 \quad \text{VCDim}\left(\{f : f(x) = \cos(ax), a \in \mathbb{R}\}\right) = \infty.$$

332 However, functions in  $\{f : f(x) = \cos(ax), a \in \mathbb{R}\}$  cannot approximate Hölder  
 333 continuous functions well.

334 (ii) A large VC-dimension is necessary for a good approximation rate, because the  
 335 best possible approximation rate is controlled by an expression of VC-dimension,  
 336 as shown in Theorem 2.4. It is shown in Theorem 6 and 8 of [13] that the VC-  
 337 dimension of ReLU networks has two types of upper bounds:  $\mathcal{O}(WL \ln W)$  and  
 338  $\mathcal{O}(WU)$ . Here,  $W$ ,  $L$ , and  $U$  are the numbers of parameters, layers, and neurons,  
 339 respectively. If we let  $N$  denote the maximum width of the network, then  $W =$   
 340  $\mathcal{O}(N^2L)$  and  $U = \mathcal{O}(NL)$ , implying that

$$341 \quad WL \ln W = \mathcal{O}(N^2L \cdot L \ln(N^2L)) = \mathcal{O}(N^2L^2 \ln(NL))$$

342 and

$$343 \quad WU = \mathcal{O}(N^2L \cdot NL) = \mathcal{O}(N^3L^2).$$

344 It follows that

$$345 \quad \text{VCDim}(\mathcal{NN}(\text{width} \leq N; \text{depth} \leq L)) \leq \min\left\{\mathcal{O}(N^2L^2 \ln(NL)), \mathcal{O}(N^3L^2)\right\},$$

deducing

$$\underbrace{C_1(\alpha, d) \left( \min\{N^2 L^2 \ln(NL), N^3 L^2\} \right)^{-\alpha/d}}_{\text{implied by Theorem 2.4}} \leq \mathcal{E}_{\alpha, d}(N, L) \leq \underbrace{C_2(\alpha, d) \left( N^2 L^2 \ln N \right)^{-\alpha/d}}_{\text{implied by Corollary 1.2 and 1.3}}, \quad (2.3)$$

where  $C_1(\alpha, d)$  and  $C_2(\alpha, d)$  are two positive constants determined by  $s, d$ , and  $C_2(s, d)$  can be explicitly expressed.

- When  $L = L_0$  is fixed, Equation (2.3) implies

$$C_1(\alpha, d, L_0) (N^2 \ln N)^{-\alpha/d} \leq \mathcal{E}_{\alpha, d}(N, L_0) \leq C_2(\alpha, d, L_0) (N^2 \ln N)^{-\alpha/d},$$

where  $C_1(\alpha, d, L_0)$  and  $C_2(\alpha, d, L_0)$  are two positive constants determined by  $\alpha, d, L_0$ .

- When  $N = N_0$  is fixed, Equation (2.3) implies

$$C_1(\alpha, d, N_0) L^{-2\alpha/d} \leq \mathcal{E}_{\alpha, d}(N_0, L) \leq C_2(\alpha, d, N_0) L^{-2\alpha/d},$$

where  $C_1(\alpha, d, N_0)$  and  $C_2(\alpha, d, N_0)$  are two positive constants determined by  $\alpha, d, N_0$ .

- It is easy to verify that Equation (2.3) is tight except for the following region

$$\{(N, L) \in \mathbb{N}^2 : C_3(\alpha, d) \leq N \leq L^{C_4(\alpha, d)}\},$$

$C_3 = C_3(\alpha, d)$  and  $C_4 = C_4(\alpha, d)$  are two positive constants. See Figure 1 for an illustration for the case  $C_3 = 1000$  and  $C_4 = 1/100$ .

Finally, let us present the detailed proof of Theorem 2.4.

*Proof of Theorem 2.4.* Recall that the VC-dimension of a function set is defined as the size of the largest set of points that this class of functions can shatter. So our goal is to find a subset of  $\mathcal{F}$  to shatter  $\mathcal{O}(\varepsilon^{-d/\alpha})$  points in  $[0, 1]^d$ , which can be divided into two steps.

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

The details of these two steps can be found below.

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

We may assume  $\varepsilon \leq 2/9$  since the case  $\varepsilon > 2/9$  is trivial. In fact,  $\varepsilon > 2/9$  implies

$$\text{VCDim}(\mathcal{F}) \geq 1 \geq 1/2 \geq 2^{-d/\alpha} > (9\varepsilon)^{-d/\alpha}.$$

Let  $K = \lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor \in \mathbb{N}^+$  and divide  $[0, 1]^d$  into  $K^d$  non-overlapping sub-cubes  $\{Q_\beta\}_\beta$  as follows:

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

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

Define a function  $\zeta_Q$  on  $[0, 1]^d$  corresponding to  $Q = Q(\mathbf{x}_0, \eta) \subseteq [0, 1]^d$  such that:

- 380 •  $\zeta_Q(\mathbf{x}_0) = (\eta/2)^\alpha/2$ ;
- 381 •  $\zeta_Q(\mathbf{x}) = 0$  for any  $\mathbf{x} \notin Q \setminus \partial Q$ , where  $\partial Q$  is the boundary of  $Q$ ;
- 382 •  $\zeta_Q$  is linear on the line that connects  $\mathbf{x}_0$  and  $\mathbf{x}$  for any  $\mathbf{x} \in \partial Q$ .

383 Define

$$384 \quad \mathcal{B} := \{\chi : \chi \text{ is a map from } \{0, 1, \dots, K-1\}^d \text{ to } \{-1, 1\}\}.$$

385 For each  $\chi \in \mathcal{B}$ , we define

$$386 \quad f_\chi(\mathbf{x}) := \sum_{\beta \in \{0, 1, \dots, K-1\}^d} \chi(\beta) \zeta_{Q_\beta}(\mathbf{x}),$$

387 where  $\zeta_{Q_\beta}(\mathbf{x})$  is the associated function introduced just above. It is easy to check that  
 388  $\{f_\chi : \chi \in \mathcal{B}\} \subseteq \text{H\"older}([0, 1]^d, \alpha, 1)$  can shatter  $K^d = \mathcal{O}(\varepsilon^{-d/\alpha})$  points in  $[0, 1]^d$ .

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

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

$$391 \quad \|\phi_\chi - f_\chi\|_{L^\infty([0, 1]^d)} \leq \varepsilon + \varepsilon/81.$$

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

$$394 \quad |\phi_\chi(\mathbf{x}) - f_\chi(\mathbf{x})| \leq \frac{82}{81}\varepsilon, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}_\chi.$$

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

$$396 \quad |\phi_\chi(\mathbf{x}) - f_\chi(\mathbf{x})| \leq \frac{82}{81}\varepsilon, \quad \text{for any } \chi \in \mathcal{B} \text{ and } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}. \quad (2.4)$$

397 Since  $Q_\beta$  has a sidelength  $\frac{1}{K} = \frac{1}{\lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor}$ , we have, for each  $\beta \in \{0, 1, \dots, K-1\}^d$  and  
 398 any  $\mathbf{x} \in \frac{1}{10}Q_\beta$ <sup>①</sup>,

$$399 \quad |f_\chi(\mathbf{x})| = |\zeta_{Q_\beta}(\mathbf{x})| \geq \frac{9}{10}|\zeta_{Q_\beta}(\mathbf{x}_{Q_\beta})| = \frac{9}{10}\left(\frac{1}{2\lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor}\right)^\alpha/2 \geq \frac{81}{80}\varepsilon, \quad (2.5)$$

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

401 Note that  $(\frac{1}{10}Q_\beta) \setminus \mathcal{H}$  is not empty, since  $\mu((\frac{1}{10}Q_\beta) \setminus \mathcal{H}) > 0$  for each  $\beta \in \{0, 1, \dots, K-1\}^d$ . Together with Equation (2.4) and (2.5), there exists  $\mathbf{x}_\beta \in (\frac{1}{10}Q_\beta) \setminus \mathcal{H}$  such that, for  
 402 each  $\beta \in \{0, 1, \dots, K-1\}^d$  and each  $\chi \in \mathcal{B}$ ,

$$404 \quad |f_\chi(\mathbf{x}_\beta)| \geq \frac{81}{80}\varepsilon > \frac{82}{81}\varepsilon \geq |f_\chi(\mathbf{x}_\beta) - \phi_\chi(\mathbf{x}_\beta)|,$$

405 Hence,  $f_\chi(\mathbf{x}_\beta)$  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  
 406  $\{\mathbf{x}_\beta : \beta \in \{0, 1, \dots, K-1\}^d\}$ . Therefore,

$$408 \quad \text{VCDim}(\mathcal{F}) \geq \text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \geq K^d = \lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor^d \geq (9\varepsilon)^{-d/\alpha},$$

409 where the last inequality comes from the fact  $\lfloor x \rfloor \geq x/2 \geq x/(2^{1/\alpha})$  for any  $x \in [1, \infty)$  and  
 410  $\alpha \in (0, 1]$ . So we finish the proof.  $\square$

---

<sup>①</sup>  $\frac{1}{10}Q_\beta$  denotes the closed cube whose sidelength is 1/10 of that of  $Q_\beta$  and which shares the same center of  $Q_\beta$ .

## 2.4 Approximation in irregular domain

We extend our analysis to general continuous functions defined on any irregular bounded set in  $\mathbb{R}^d$ . The key idea is to extend the target function to a hypercube while preserving the modulus of continuity. The extension of continuous (smooth) functions has been widely studied, e.g., [39] for smooth functions and [38] for continuous functions. For simplicity, we use Lemma 4.2 of [33]. The proof can be found therein. For a general set  $E \subseteq \mathbb{R}^d$ , the modulus of continuity of  $f \in C(E)$  is defined via

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

In particular,  $\omega_f(\cdot)$  is short of  $\omega_f^E(\cdot)$  in the case of  $E = [0, 1]^d$ . Then, Theorem 1.1 can be generalized to  $f \in C(E)$  for any bounded set  $E \subseteq [-R, R]^d$  with  $R > 0$ , as shown in the following theorem.

**Theorem 2.5.** *Given any bounded continuous function  $f \in C(E)$  with  $E \subseteq [-R, R]^d$  and  $R > 0$ , for any  $N \in \mathbb{N}^+$ ,  $L \in \mathbb{N}^+$ , and  $p \in [1, \infty]$ , there exists a function  $\phi$  implemented by a ReLU network with width  $C_1 \max \{d \lfloor N^{1/d} \rfloor, N + 2\}$  and depth  $11L + C_2$  such that*

$$\|f - \phi\|_{L^p(E)} \leq 131(2R)^{d/p} \sqrt{d} \omega_f^E \left( 2R \left( N^2 L^2 \log_3(N + 2) \right)^{-1/d} \right),$$

where  $C_1 = 16$  and  $C_2 = 18$  if  $p \in [1, \infty)$ ;  $C_1 = 3^{d+3}$  and  $C_2 = 18 + 2d$  if  $p = \infty$ .

*Proof.* Given any bounded continuous function  $f \in C(E)$ , by Lemma 4.2 of [33] via setting  $S = [-R, R]^d$ , there exists  $g \in C([-R, R]^d)$  such that

- $g(\mathbf{x}) = f(\mathbf{x})$  for any  $\mathbf{x} \in E \subseteq S = [-R, R]^d$ ;
- $\omega_g^S(r) = \omega_f^E(r)$  for any  $r \geq 0$ .

Define

$$\tilde{g}(\mathbf{x}) := g(2R\mathbf{x} - R), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

By applying Theorem 1.1 to  $\tilde{g} \in C([0, 1]^d)$ , there exists a function  $\tilde{\phi}$  implemented by a ReLU network with width  $C_1 \max \{d \lfloor N^{1/d} \rfloor, N + 2\}$  and depth  $11L + C_2$  such that

$$\|\tilde{\phi} - \tilde{g}\|_{L^p([0, 1]^d)} \leq 131 \sqrt{d} \omega_{\tilde{g}} \left( \left( N^2 L^2 \log_3(N + 2) \right)^{-1/d} \right),$$

where  $C_1 = 16$  and  $C_2 = 18$  if  $p \in [1, \infty)$ ;  $C_1 = 3^{d+3}$  and  $C_2 = 18 + 2d$  if  $p = \infty$ .

Note that  $f(\mathbf{x}) = g(\mathbf{x}) = \tilde{g}(\frac{\mathbf{x}+R}{2R})$  for any  $\mathbf{x} \in E \subseteq S = [-R, R]^d$  and

$$\omega_{\tilde{g}}(r) = \omega_g^S(2Rr) = \omega_f^E(2Rr), \quad \text{for any } r \geq 0.$$

Define  $\phi(\mathbf{x}) := \tilde{\phi}(\frac{\mathbf{x}+R}{2R}) = \tilde{\phi} \circ \mathcal{L}(\mathbf{x})$  for any  $\mathbf{x} \in \mathbb{R}^d$ , where  $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is an affine linear map given by  $\mathcal{L}(\mathbf{x}) = \frac{\mathbf{x}+R}{2R}$ . Clearly,  $\phi$  can be implemented by a ReLU network with width  $C_1 \max \{d \lfloor N^{1/d} \rfloor, N + 2\}$  and depth  $11L + C_2$ , where  $C_1 = 16$  and  $C_2 = 18$  if  $p \in [1, \infty)$ ;

442  $C_1 = 3^{d+3}$  and  $C_2 = 18 + 2d$  if  $p = \infty$ . Moreover, for any  $\mathbf{x} \in E \subseteq S = [-R, R]^d$ , we have  
 443  $\frac{\mathbf{x}+R}{2R} \in [0, 1]^d$ , implying

$$\begin{aligned}
 \|\phi - f\|_{L^p(E)} &= \|\phi - g\|_{L^p(E)} = \|\tilde{\phi} \circ \mathcal{L} - \tilde{g} \circ \mathcal{L}\|_{L^p(E)} \\
 &\leq \|\tilde{\phi} \circ \mathcal{L} - \tilde{g} \circ \mathcal{L}\|_{L^p([-R, R]^d)} = (2R)^{d/p} \|\tilde{\phi} - \tilde{g}\|_{L^p([0, 1]^d)} \\
 444 &\leq 131(2R)^{d/p} \sqrt{d} \omega_{\tilde{g}} \left( (N^2 L^2 \log_3(N+2))^{-1/d} \right) \\
 &= 131(2R)^{d/p} \sqrt{d} \omega_f^E \left( 2R(N^2 L^2 \log_3(N+2))^{-1/d} \right).
 \end{aligned}$$

445 With the discussion above, we have proved Theorem 2.5. □

### 446 3 Proof of Theorem 2.1

447 We will prove Theorem 2.1 in this section. We first present the key ideas in Sec-  
 448 tion 3.1. The detailed proof is presented in Section 3.3, based on two propositions in  
 449 Section 3.1, the proofs of which can be found in Section 4.

#### 450 3.1 Key ideas of proving Theorem 2.1

451 Given an arbitrary  $f \in C([0, 1]^d)$ , our goal is to construct an almost piecewise  
 452 constant function  $\phi$  implemented by a ReLU network to approximate  $f$  well. To this end,  
 453 we introduce a piecewise constant function  $f_p \approx f$  serving as an intermediate approximant  
 454 in our construction in the sense that

$$455 \quad f \approx f_p \text{ on } [0, 1]^d \quad \text{and} \quad f_p \approx \phi \text{ on } [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta).$$

456 The approximation in  $f \approx f_p$  is a simple and standard technique in constructive approx-  
 457 imation. The most technical part is to design a ReLU network with the desired width  
 458 and depth to implement a function  $\phi$  with  $\phi \approx f_p$  outside  $\Omega([0, 1]^d, K, \delta)$ . See Figure 4  
 459 for an illustration. The introduction of the trifling region is to ease the construction  
 460 of  $\phi$ , which is a continuous piecewise linear function, to approximate the discontinuous  
 461 function  $f_p$  by removing the difficulty near discontinuous points, essentially smoothing  
 462  $f_p$  by restricting the approximation domain in  $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ .

463 Now let us discuss the detailed steps of construction.

464 (i) First, divide  $[0, 1]^d$  into a union of important regions  $\{Q_{\beta}\}_{\beta}$  and the trifling re-  
 465 gion  $\Omega([0, 1]^d, K, \delta)$ , where each  $Q_{\beta}$  is associated with a representative  $\mathbf{x}_{\beta} \in Q_{\beta}$   
 466 such that  $f(\mathbf{x}_{\beta}) = f_p(\mathbf{x}_{\beta})$  for each index vector  $\beta \in \{0, 1, \dots, K-1\}^d$ , where  
 467  $K = \mathcal{O}((N^2 L^2 \ln N)^{1/d})$  is the partition number per dimension (see Figure 7 for  
 468 examples for  $d = 1$  and  $d = 2$ ).

469 (ii) Next, we design a vector function  $\Phi_1(\mathbf{x})$  constructed via

$$470 \quad \Phi_1(\mathbf{x}) = [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T$$

471 to project the whole cube  $Q_{\beta}$  to a  $d$ -dimensional index  $\beta$  for each  $\beta$ , where each  
 472 one-dimensional function  $\phi_1$  is a step function implemented by a ReLU network.



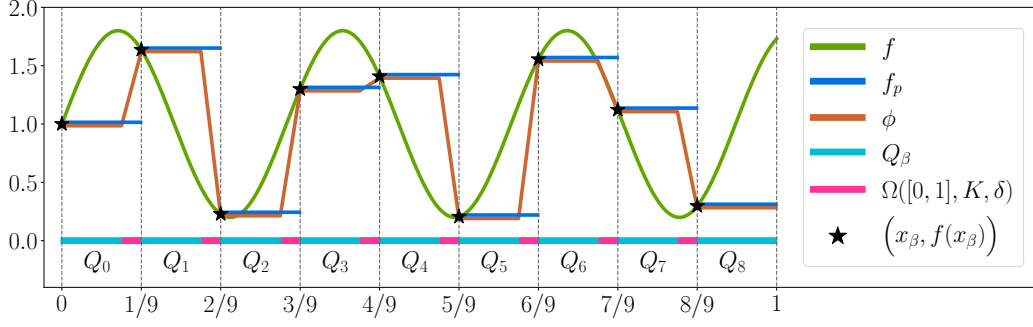


Figure 4: An illustration of  $f$ ,  $f_p$ ,  $\phi$ ,  $x_\beta$ ,  $Q_\beta$ , and the trifling region  $\Omega([0, 1]^d, K, \delta)$  in the one-dimensional case for  $\beta \in \{0, 1, \dots, K-1\}^d$ , where  $K = N^2 L^2 \log_3(N+2)$  and  $d = 1$  with  $N = 1$  and  $L = 3$ .  $f$  is the target function;  $f_p$  is the piecewise constant function approximating  $f$ ;  $\phi$  is a function, implemented by a ReLU network, approximating  $f$ ; and  $x_\beta$  is a representative of  $Q_\beta$ . The measure of  $\Omega([0, 1]^d, K, \delta)$  can be arbitrarily small as we shall see in the proof of Theorem 1.1.

(iii) The third step is to solve a point fitting problem. To be precise, we construct a function  $\phi_2$  implemented by a ReLU network to map  $\beta \in \{0, 1, \dots, K-1\}^d$  approximately to  $f_p(x_\beta) = f(x_\beta)$ . Then  $\phi_2 \circ \Phi_1(x) = \phi_2(\beta) \approx f_p(x_\beta) = f(x_\beta) \approx f(x)$  for any  $x \in Q_\beta$  and each  $\beta$ , implying  $\phi := \phi_2 \circ \Phi_1 \approx f_p \approx f$  on  $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ . We would like to point out that we only need to care about the values of  $\phi_2$  at a set of points  $\{0, 1, \dots, K-1\}^d$  in the construction of  $\phi_2$  according to our design  $\phi = \phi_2 \circ \Phi_1$  as illustrated in Figure 5. Therefore, it is not necessary to care about the values of  $\phi_2$  sampled outside the set  $\{0, 1, \dots, K-1\}^d$ , which is a key point to ease the design of a ReLU network to implement  $\phi_2$  as we shall see later.

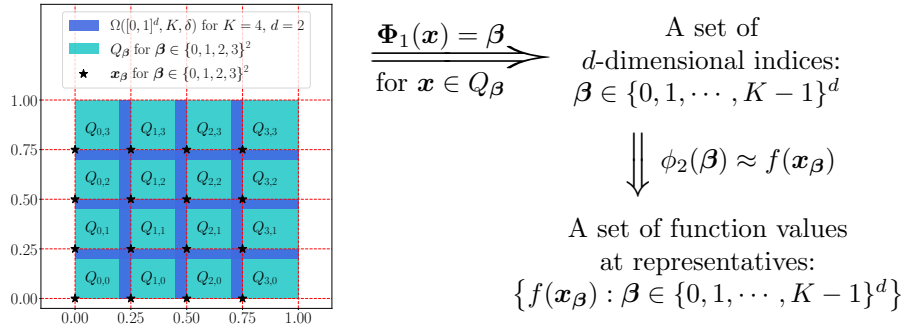


Figure 5: An illustration of the desired function  $\phi = \phi_2 \circ \Phi_1$ . Note that  $\phi \approx f$  on  $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ , since  $\phi(x) = \phi_2 \circ \Phi_1(x) = \phi_2(\beta) \approx f(x_\beta) \approx f(x)$  for any  $x \in Q_\beta$  and each  $\beta \in \{0, 1, \dots, K-1\}^d$ .

We remark that in Figure 5, we have

$$\phi(x) = \phi_2 \circ \Phi_1(x) = \phi_2(\beta) \stackrel{\mathcal{E}_1}{\approx} f(x_\beta) \stackrel{\mathcal{E}_2}{\approx} f(x)$$

for any  $x \in Q_\beta$  and each  $\beta \in \{0, 1, \dots, K-1\}^d$ . Thus,  $\phi - f$  is bounded by  $\mathcal{E}_1 + \mathcal{E}_2$  outside the trifling region. Observe that  $\mathcal{E}_2$  is bounded by  $\omega_f(\sqrt{d}/K)$ . As we shall see later in

Section 3.3,  $\mathcal{E}_1$  can also be bounded by  $\omega_f(\sqrt{d}/K)$  by applying Proposition 3.2. Hence,  $\phi - f$  is controlled by  $2\omega_f(\sqrt{d}/K)$  outside the trifling region, which deduces the desired approximation error.

Finally, we discuss how to implement  $\Phi_1$  and  $\phi_2$  by deep ReLU networks with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  using two propositions as we shall prove in Section 4.2 and 4.3 later. We first show how to construct a ReLU network with the desired width and depth by Proposition 3.1 to implement a one-dimensional step function  $\phi_1$ . Then  $\Phi_1$  can be attained via defining

$$\Phi_1(\mathbf{x}) = [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d.$$

**Proposition 3.1.** *For any  $N, L, d \in \mathbb{N}^+$  and  $\delta \in (0, \frac{1}{3K}]$  with*

$$K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor \lfloor n^{1/d} \rfloor, \quad \text{where } n = \lfloor \log_3(N+2) \rfloor,$$

*there exists a one-dimensional function  $\phi$  implemented by a ReLU network with width  $8\lfloor N^{1/d} \rfloor + 3$  and depth  $2\lfloor L^{1/d} \rfloor + 5$  such that*

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

The setting  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor = \mathcal{O}(N^{2/d} L^{2/d} n^{1/d})$  is not neat here, but it is very convenient for later use. The construction of  $\phi_2$  is a direct result of Proposition 3.2 below, the proof of which relies on the bit extraction technique in [3].

**Proposition 3.2.** *Given any  $\varepsilon > 0$  and arbitrary  $N, L, J \in \mathbb{N}^+$  with  $J \leq N^2 L^2 \lfloor \log_3(N+2) \rfloor$ , assume  $y_j \geq 0$  for  $j = 0, 1, \dots, J-1$  are samples with*

$$|y_j - y_{j-1}| \leq \varepsilon, \quad \text{for } j = 1, 2, \dots, J-1.$$

*Then there exists  $\phi \in \mathcal{NN}(\#input = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6L + 10; \#output = 1)$  such that*

$$(i) \quad |\phi(j) - y_j| \leq \varepsilon \text{ for } j = 0, 1, \dots, J-1.$$

$$(ii) \quad 0 \leq \phi(x) \leq \max\{y_j : j = 0, 1, \dots, J-1\} \text{ for any } x \in \mathbb{R}.$$

## 3.2 Construction of final network

We will discuss the construction of the final network approximating the target function with the same setting as in Section 3.1. There are two main parts: 1) Construct the final network architecture based on Proposition 3.1 and 3.2; 2) Implement the network architectures in Proposition 3.1 and 3.2.

### Final network architecture based on Proposition 3.1 and 3.2

By the idea mentioned in Figure 5, the final network architecture can be implemented as shown in Figure 6.

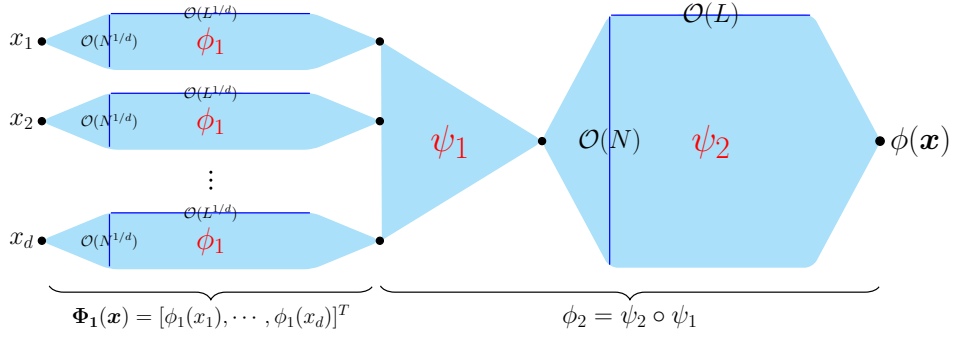


Figure 6: An illustration of the final network architecture with width  $\max\{\mathcal{O}(dN^{1/d}), \mathcal{O}(N)\}$  and depth  $\mathcal{O}(L)$ .  $\psi_1 : \mathbb{R}^d \rightarrow \mathbb{R}$  is a linear function.  $\phi_1$  and  $\psi_2$  are implemented via Proposition 3.1 and 3.2, respectively.

Note that  $\phi_1$  in Figure 6 is a step function mapping  $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \leq K-1\}}]$  to  $k$  for each  $k \in \{0, 1, \dots, K-1\}$ . It can be easily implemented via Proposition 3.1. Clearly, by defining  $\Phi_1(\mathbf{x}) = [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T$ ,  $\Phi_1$  maps  $\mathbf{x} \in Q_\beta$  to  $\beta$ .

As shown in Figure 5, we need to design a network to compute  $\phi_2$  mapping  $\beta \in \{0, 1, \dots, K-1\}^d$  approximately to  $f(\mathbf{x}_\beta)$ . To this end, we first construct a **linear** function  $\psi_1 : \mathbb{R}^d \rightarrow \mathbb{R}$  mapping  $\beta \in \{0, 1, \dots, K-1\}^d$  to  $\mathbb{R}$  for the purpose of converting a  $d$ -dimensional point-fitting problem to a one-dimensional one, and then construct a network to compute  $\psi_2$  with  $\psi_2(\psi_1(\beta)) \approx f(\mathbf{x}_\beta)$  via applying Proposition 3.2. Thus, we have  $\phi_2(\beta) := \psi_2 \circ \psi_1(\beta) \approx f(\mathbf{x}_\beta)$  as desired.

## Network architectures in Proposition 3.1 and 3.2

To prove Proposition 3.1, we need to construct a ReLU network with width  $\mathcal{O}(N^{1/d})$  and depth  $\mathcal{O}(L^{1/d})$  to compute a step function with  $\mathcal{O}((N^2 L^2 \ln N)^{1/d})$  “steps” outside the trifling region. It is easy to construct a ReLU network with  $\mathcal{O}(W)$  parameters to compute a step function with  $W$  “steps” outside a small region. As we shall see later in Section 4.2, the composition architecture of ReLU networks can help to implement step functions with much more “steps”. Refer to Section 4.2 for the detailed proof of Proposition 3.1.

Proposition 3.2 essentially solves a point-fitting problem with  $N^2 L^2 \lceil \log_3(N+2) \rceil$  points via a ReLU network with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$ . Set  $M = N^2 L$ ,  $\widehat{L} = \lceil \log_3(N+2) \rceil$ , and represent  $j \in \{0, 1, \dots, M\widehat{L}-1\}$  via  $j = m\widehat{L} + k$ , where  $m \in \{0, 1, \dots, M-1\}$  and  $k \in \{0, 1, \dots, \widehat{L}-1\}$ .

Define  $a_{m,k} = \lfloor y_{m,k}/\varepsilon \rfloor$  where  $y_{m,k} = y_{m\widehat{L}+k}$ . Then

$$|a_{m,k}\varepsilon - y_{m,k}| = |\lfloor y_{m,k}/\varepsilon \rfloor \varepsilon - y_{m,k}| \leq \varepsilon.$$

It suffices to prove  $\phi(m,k) = a_{m,k}$ . The assumption  $|y_j - y_{j-1}| < \varepsilon$  implies that  $b_{m,k} := a_{m,k} - a_{m,k-1} \in \{-1, 0, 1\}$ . Thus, there exist  $c_{m,k} \in \{0, 1\}$  and  $d_{m,k} \in \{0, 1\}$  such that  $b_{m,k} = c_{m,k} - d_{m,k}$ .

Note that

$$a_{m,k} = a_{m,0} + \sum_{j=1}^k (a_{m,j} - a_{m,j-1}) = a_{m,0} + \sum_{j=1}^k b_{m,j} = a_{m,0} + \sum_{j=1}^k c_{m,j} - \sum_{j=1}^k d_{m,j}.$$

It is easy to construct a ReLU network with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  ( $\mathcal{O}(N^2L)$  parameters in total) to compute  $\phi_1$  such that  $\phi_1(m) = a_{m,0}$  for each  $m \in \{0, 1, \dots, M-1\}$  with  $M = N^2L$ . By the bit extraction technique in [3], one could construct  $\phi_2, \phi_3 \in \mathcal{NN}(\text{width} \leq \mathcal{O}(N); \text{depth} \leq \mathcal{O}(L))$  such that  $\phi_2(m, k) = \sum_{j=1}^k c_{m,j}$  and  $\phi_3(m, k) = \sum_{j=1}^k d_{m,j}$ . Thus,  $\phi(m, k) := \phi_1(m) + \phi_2(m, k) - \phi_3(m, k) = a_{m,k}$  as desired.

In order to use the bit extraction technique (two types of bits 0 or 1) to solve the point-fitting problem, we essentially simplify the target as discussed above. That is,

$$\begin{aligned} \text{positive number } y_{m,k} &\longrightarrow \text{integer } a_{m,k} = \lfloor y_{m,k}/\varepsilon \rfloor \stackrel{\varepsilon}{\approx} y_{m,k} \\ &\longrightarrow b_{m,k} = a_{m,k} - a_{m,k-1} \in \{-1, 0, 1\} \\ &\longrightarrow b_{m,k} = c_{m,k} - d_{m,k} \text{ with } c_{m,k}, d_{m,k} \in \{0, 1\}. \end{aligned}$$

The detailed proof of Proposition 3.2 can be found in Section 4.3.

### 3.3 Detailed proof

We essentially construct an almost piecewise constant function implemented by a ReLU network with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  to approximate  $f$ . We may assume  $f$  is not a constant function since it is a trivial case. Then  $\omega_f(r) > 0$  for any  $r > 0$ . It is clear that  $|f(\mathbf{x}) - f(\mathbf{0})| \leq \omega_f(\sqrt{d})$  for any  $\mathbf{x} \in [0, 1]^d$ . Define  $\tilde{f} = f - f(\mathbf{0}) + \omega_f(\sqrt{d})$ , then  $0 \leq \tilde{f}(\mathbf{x}) \leq 2\omega_f(\sqrt{d})$  for any  $\mathbf{x} \in [0, 1]^d$ .

Let  $M = N^2L$ ,  $n = \lfloor \log_3(N+2) \rfloor$ ,  $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor$ , and  $\delta$  be an arbitrary number in  $(0, \frac{1}{3K}]$ . The proof can be divided into four steps as follows:

1. Normalize  $f$  as  $\tilde{f}$ , divide  $[0, 1]^d$  into a union of sub-cubes  $\{Q_\beta\}_{\beta \in \{0, 1, \dots, K-1\}^d}$  and the trifling region  $\Omega([0, 1]^d, K, \delta)$ , and denote  $\mathbf{x}_\beta$  as the vertex of  $Q_\beta$  with minimum  $\|\cdot\|_1$  norm;
2. Construct a sub-network to implement a vector function  $\Phi_1$  projecting the whole cube  $Q_\beta$  to the  $d$ -dimensional index  $\beta$  for each  $\beta$ , i.e.,  $\Phi_1(\mathbf{x}) = \beta$  for all  $\mathbf{x} \in Q_\beta$ ;
3. Construct a sub-network to implement a function  $\phi_2$  mapping the index  $\beta$  approximately to  $\tilde{f}(\mathbf{x}_\beta)$ . This core step can be further divided into three sub-steps:
  - 3.1. Construct a sub-network to implement  $\psi_1$  bijectively mapping the index set  $\{0, 1, \dots, K-1\}^d$  to an auxiliary set  $\mathcal{A}_1 \subseteq \{\frac{j}{2K^d} : j = 0, 1, \dots, 2K^d\}$  defined later (see Figure 8 for an illustration);
  - 3.2. Determine a continuous piecewise linear function  $g$  with a set of breakpoints  $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$  satisfying: 1) assign the values of  $g$  at breakpoints in  $\mathcal{A}_1$  based on  $\{\tilde{f}(\mathbf{x}_\beta)\}_\beta$ , i.e.,  $g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$ ; 2) assign the values of  $g$  at breakpoints in  $\mathcal{A}_2 \cup \{1\}$  to reduce the variation of  $g$  for applying Proposition 3.2;
  - 3.3. Apply Proposition 3.2 to construct a sub-network to implement a function  $\psi_2$  approximating  $g$  well on  $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ . Then the desired function  $\phi_2$  is given by  $\phi_2 = \psi_2 \circ \psi_1$  satisfying  $\phi_2(\beta) = \psi_2 \circ \psi_1(\beta) \approx g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$ ;
4. Construct the final network to implement the desired function  $\phi$  such that  $\phi(\mathbf{x}) = \phi_2 \circ \Phi_1(\mathbf{x}) + f(\mathbf{0}) - \omega_f(\sqrt{d}) \approx \tilde{f}(\mathbf{x}_\beta) + f(\mathbf{0}) - \omega_f(\sqrt{d}) = f(\mathbf{x}_\beta) \approx f(\mathbf{x})$  for any  $\mathbf{x} \in Q_\beta$  and  $\beta \in \{0, 1, \dots, K-1\}^d$ .

583 The details of these steps can be found below.

584 **Step 1:** Divide  $[0, 1]^d$  into  $\{Q_\beta\}_{\beta \in \{0, 1, \dots, K-1\}^d}$  and  $\Omega([0, 1]^d, K, \delta)$ .

585 Define  $\mathbf{x}_\beta := \beta/K$  and

586 
$$Q_\beta := \left\{ \mathbf{x} = [x_1, \dots, x_d]^T \in [0, 1]^d : x_i \in \left[ \frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot \mathbb{1}_{\{\beta_i \leq K-2\}} \right], i = 1, \dots, d \right\}$$

587 for each  $d$ -dimensional index  $\beta = [\beta_1, \dots, \beta_d]^T \in \{0, 1, \dots, K-1\}^d$ . Recall that  $\Omega([0, 1]^d, K, \delta)$   
 588 is the trifling region defined in Equation (2.1). Apparently,  $\mathbf{x}_\beta$  is the vertex of  $Q_\beta$  with  
 589 minimum  $\|\cdot\|_1$  norm and

590 
$$[0, 1]^d = \left( \cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta \right) \cup \Omega([0, 1]^d, K, \delta).$$

591 See Figure 7 for illustrations.

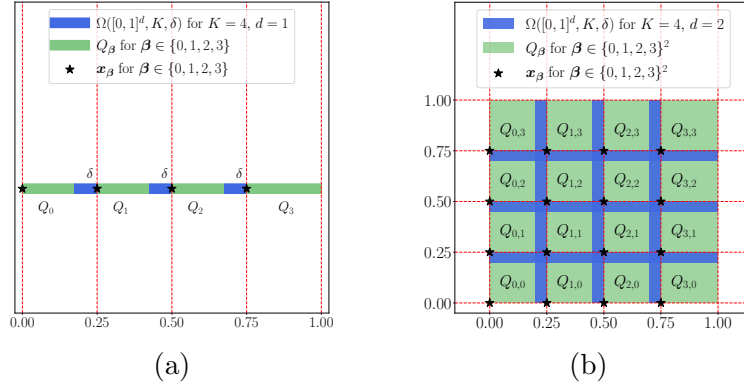


Figure 7: Illustrations of  $\Omega([0, 1]^d, K, \delta)$ ,  $Q_\beta$ , and  $\mathbf{x}_\beta$  for  $\beta \in \{0, 1, \dots, K-1\}^d$ . (a)  $K = 4$  and  $d = 1$ . (b)  $K = 4$  and  $d = 2$ .

592 **Step 2:** Construct  $\Phi_1$  mapping  $\mathbf{x} \in Q_\beta$  to  $\beta$ .

593 By Proposition 3.1, there exists  $\phi_1 \in \mathcal{NN}(\text{width} \leq 8\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 2\lfloor L^{1/d} \rfloor + 5)$   
 594 such that

595 
$$\phi_1(x) = k, \quad \text{if } x \in \left[ \frac{k}{K}, \frac{k+1}{K} - \delta \cdot \mathbb{1}_{\{k \leq K-2\}} \right] \text{ for } k = 0, 1, \dots, K-1.$$

596 It follows that  $\phi_1(x_i) = \beta_i$  if  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in Q_\beta$  for each  $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T$ .

597 By defining

598 
$$\Phi_1(\mathbf{x}) := [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d,$$

599 we have  $\Phi_1(\mathbf{x}) = \beta$  if  $\mathbf{x} \in Q_\beta$  for each  $\beta \in \{0, 1, \dots, K-1\}^d$ .

600 **Step 3:** Construct  $\phi_2$  mapping  $\beta$  approximately to  $\tilde{f}(\mathbf{x}_\beta)$ .

601 The construction of the sub-network implementing  $\phi_2$  is essentially based on Propo-  
 602 sition 3.2. To meet the requirements of applying Proposition 3.2, we first define two  
 603 auxiliary set  $\mathcal{A}_1$  and  $\mathcal{A}_2$  as

604 
$$\mathcal{A}_1 := \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\}$$

605 and

$$606 \quad \mathcal{A}_2 := \left\{ \frac{i}{K^{d-1}} + \frac{K+k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\}.$$

607 Clearly,  $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\} = \left\{ \frac{j}{2K^d} : j = 0, 1, \dots, 2K^d \right\}$  and  $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ . See Figure 7 for an  
608 illustration of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . Next, we further divide this step into three sub-steps.

609 **Step 3.1:** Construct  $\psi_1$  bijectively mapping  $\{0, 1, \dots, K-1\}^d$  to  $\mathcal{A}_1$ .

610 Inspired by the binary representation, we define

$$611 \quad \psi_1(\mathbf{x}) := \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d. \quad (3.1)$$

612 Then  $\psi_1$  is a linear function bijectively mapping the index set  $\{0, 1, \dots, K-1\}^d$  to

$$613 \quad \begin{aligned} & \left\{ \frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i} : \beta \in \{0, 1, \dots, K-1\}^d \right\} \\ & = \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\} = \mathcal{A}_1. \end{aligned}$$

614 **Step 3.2:** Construct  $g$  to satisfy  $g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$  and to meet the requirements of  
615 applying Proposition 3.2.

616 Let  $g : [0, 1] \rightarrow \mathbb{R}$  be a continuous piecewise linear function with a set of breakpoints  
617  $\left\{ \frac{j}{2K^d} : j = 0, 1, \dots, 2K^d \right\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$  and the values of  $g$  at these breakpoints satisfy  
618 the following properties:

- 619 • The values of  $g$  at the breakpoints in  $\mathcal{A}_1$  are set as

$$620 \quad g(\psi_1(\beta)) = \tilde{f}(\mathbf{x}_\beta), \quad \text{for any } \beta \in \{0, 1, \dots, K-1\}^d; \quad (3.2)$$

- 621 • At the breakpoint 1, let  $g(1) = \tilde{f}(\mathbf{1})$ , where  $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathbb{R}^d$ ;
- 622 • The values of  $g$  at the breakpoints in  $\mathcal{A}_2$  are assigned to reduce the variation of  $g$ ,  
623 which is a requirement of applying Proposition 3.2. Note that

$$624 \quad \left\{ \frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}} \right\} \subseteq \mathcal{A}_1 \cup \{1\}, \quad \text{for } i = 1, 2, \dots, K^{d-1},$$

625 implying the values of  $g$  at  $\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}$  and  $\frac{i}{K^{d-1}}$  have been assigned for  $i = 1, 2, \dots, K^{d-1}$ .  
626 Thus, the values of  $g$  at the breakpoints in  $\mathcal{A}_2$  can be successfully assigned by  
627 letting  $g$  linear on each interval  $\left[ \frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}} \right]$  for  $i = 1, 2, \dots, K^{d-1}$ , since  
628  $\mathcal{A}_2 \subseteq \bigcup_{i=1}^{K^{d-1}} \left[ \frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}} \right]$ . See Figure 8 for an illustration.

629 Apparently, such a function  $g$  exists (see Figure 8 for an example) and satisfies

$$630 \quad \left| g\left(\frac{j}{2K^d}\right) - g\left(\frac{j-1}{2K^d}\right) \right| \leq \max \left\{ \omega_f\left(\frac{1}{K}\right), \omega_f(\sqrt{d})/K \right\} \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \quad \text{for } j = 1, 2, \dots, 2K^d,$$

631 and

$$632 \quad 0 \leq g\left(\frac{j}{2K^d}\right) \leq 2\omega_f(\sqrt{d}), \quad \text{for } j = 0, 1, \dots, 2K^d.$$

633 **Step 3.3:** Construct  $\psi_2$  approximating  $g$  well on  $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ .

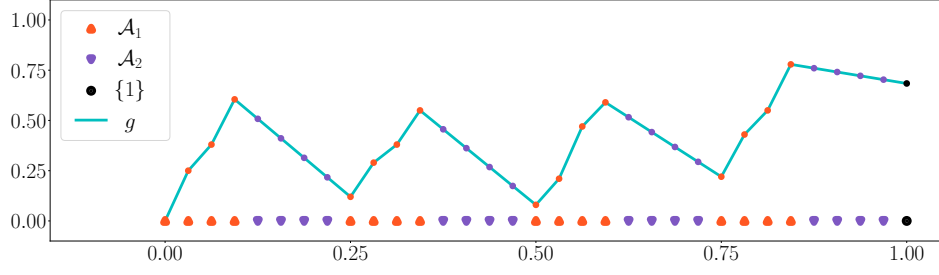


Figure 8: An illustration of  $\mathcal{A}_1$ ,  $\mathcal{A}_2$ ,  $\{1\}$ , and  $g$  for  $d = 2$  and  $K = 4$ .

Note that

$$2K^d = 2(\lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor)^d \leq 2(N^2 L^2 n) \leq N^2 \lceil \sqrt{2}L \rceil^2 \lceil \log_3(N+2) \rceil.$$

By Proposition 3.2 (set  $y_j = g(\frac{j}{2K^d})$  and  $\varepsilon = \omega_f(\frac{\sqrt{d}}{K}) > 0$  therein), there exists

$$\tilde{\psi}_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6\lceil \sqrt{2}L \rceil + 10; \#output = 1)$$

such that

$$|\tilde{\psi}_2(j) - g(\frac{j}{2K^d})| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad \text{for } j = 0, 1, \dots, 2K^d - 1,$$

and

$$0 \leq \tilde{\psi}_2(x) \leq \max\{g(\frac{j}{2K^d}) : j = 0, 1, \dots, 2K^d - 1\} \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}.$$

By defining  $\psi_2(x) := \tilde{\psi}_2(2K^d x)$  for any  $x \in \mathbb{R}$ , we have  $\psi_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6\lceil \sqrt{2}L \rceil + 10; \#output = 1)$ ,

$$0 \leq \psi_2(x) = \tilde{\psi}_2(2K^d x) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}, \quad (3.3)$$

and

$$|\psi_2(\frac{j}{2K^d}) - g(\frac{j}{2K^d})| = |\tilde{\psi}_2(j) - g(\frac{j}{2K^d})| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad \text{for } j = 0, 1, \dots, 2K^d - 1. \quad (3.4)$$

Let us end Step 3 by defining the desired function  $\phi_2$  as  $\phi_2 := \psi_2 \circ \psi_1$ . Note that  $\psi_1 : \mathbb{R}^d \rightarrow \mathbb{R}$  is a linear function and  $\psi_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6\lceil \sqrt{2}L \rceil + 10; \#output = 1)$ . Thus,  $\phi_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6\lceil \sqrt{2}L \rceil + 10; \#output = 1)$ . By Equation (3.2) and (3.4), we have

$$|\phi_2(\beta) - \tilde{f}(\mathbf{x}_\beta)| = |\psi_2(\psi_1(\beta)) - g(\psi_1(\beta))| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad (3.5)$$

for any  $\beta \in \{0, 1, \dots, K-1\}^d$ . Equation (3.3) and  $\phi_2 = \psi_2 \circ \psi_1$  implies

$$0 \leq \phi_2(\mathbf{x}) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d. \quad (3.6)$$

**Step 4:** Construct the final network to implement the desired function  $\phi$ .

Define  $\phi := \phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$ . Since  $\phi_1 \in \mathcal{NN}(\text{width} \leq 8\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 2\lfloor L^{1/d} \rfloor + 5)$ , we have  $\Phi_1 \in \mathcal{NN}(\#input = d; \text{width} \leq 8d\lfloor N^{1/d} \rfloor + 3d; \text{depth} \leq 2L + 10)$ .



5; #output =  $d$ ). It follows from the fact  $\lceil \sqrt{2}L \rceil \leq \lceil \frac{3}{2}L \rceil \leq \frac{3}{2}L + \frac{1}{2}$  that  $6\lceil \sqrt{2}L \rceil + 10 \leq 9L + 13$ ,  
implying

$$\begin{aligned} \phi_2 &\in \mathcal{NN}(\text{\#input} = 1; \text{width} \leq 16N + 30; \text{depth} \leq 6\lceil \sqrt{2}L \rceil + 10; \text{\#output} = 1) \\ &\subseteq \mathcal{NN}(\text{\#input} = 1; \text{width} \leq 16N + 30; \text{depth} \leq 9L + 13; \text{\#output} = 1). \end{aligned}$$

Thus,  $\phi = \phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$  is in

$$\mathcal{NN}(\text{width} \leq \max\{8d\lfloor N^{1/d} \rfloor + 3d, 16N + 30\}; \text{depth} \leq (2L + 5) + (9L + 13) = 11L + 18).$$

Now let us estimate the approximation error. Note that  $f = \tilde{f} + f(\mathbf{0}) - \omega_f(\sqrt{d})$ . By Equation (3.5), for any  $\mathbf{x} \in Q_\beta$  and  $\beta \in \{0, 1, \dots, K-1\}^d$ , we have

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &= |\tilde{f}(\mathbf{x}) - \phi_2(\Phi_1(\mathbf{x}))| = |\tilde{f}(\mathbf{x}) - \phi_2(\beta)| \\ &\leq |\tilde{f}(\mathbf{x}) - \tilde{f}(\mathbf{x}_\beta)| + |\tilde{f}(\mathbf{x}_\beta) - \phi_2(\beta)| \\ &\leq \omega_f\left(\frac{\sqrt{d}}{K}\right) + \omega_f\left(\frac{\sqrt{d}}{K}\right) \leq 2\omega_f\left(64\sqrt{d}\left(N^2L^2\log_3(N+2)\right)^{-1/d}\right), \end{aligned}$$

where the last inequality comes from the fact

$$K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor \geq \frac{N^{2/d}L^{2/d}n^{1/d}}{32} = \frac{N^{2/d}L^{2/d}\lfloor \log_3(N+2) \rfloor^{1/d}}{32} \geq \frac{(N^2L^2\log_3(N+2))^{1/d}}{64},$$

for any  $N, L \in \mathbb{N}^+$ . Recall the fact  $\omega_f(j \cdot r) \leq j \cdot \omega_f(r)$  for any  $j \in \mathbb{N}^+$  and  $r \in [0, \infty)$ .  
Therefore, for any  $\mathbf{x} \in \bigcup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta = [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$ , we have

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq 2\omega_f\left(64\sqrt{d}\left(N^2L^2\log_3(N+2)\right)^{-1/d}\right) \\ &\leq 2\left\lceil 64\sqrt{d} \right\rceil \omega_f\left(\left(N^2L^2\log_3(N+2)\right)^{-1/d}\right) \\ &\leq 130\sqrt{d}\omega_f\left(\left(N^2L^2\log_3(N+2)\right)^{-1/d}\right). \end{aligned}$$

It remains to show the upper bound of  $\phi$ . By Equation (3.6) and  $\phi = \phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$ , it holds that  $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$ . Thus, we finish the proof.

## 4 Proofs of propositions in Section 3.1

In this section, we will prove Proposition 3.1 and 3.2. We first introduce several basic results of ReLU networks. Next, we prove these two propositions based on these basic results.

### 4.1 Basic results of ReLU networks

To simplify the proofs of two propositions in Section 3.1, we introduce three lemmas below, which are basic results of ReLU networks

**Lemma 4.1.** *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  $\phi \in \mathcal{NN}(\text{\#input} = 1; \text{widthvec} = [2N_1, 2N_2 + 1]; \text{\#output} = 1)$  satisfying the following conditions.*

682 (i)  $\phi(x_i) = y_i$  for  $i = 0, 1, \dots, N_1(N_2 + 1)$ .

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

684 **Lemma 4.2.** *Given any  $N, L, d \in \mathbb{N}^+$ , it holds that*

$$\begin{aligned} & \mathcal{NN}(\#input = d; \text{widthvec} = [N, NL]; \#output = 1) \\ 685 & \subseteq \mathcal{NN}(\#input = d; \text{width} \leq 2N + 2; \text{depth} \leq L + 1; \#output = 1). \end{aligned}$$

686 **Lemma 4.3.** *For any  $n \in \mathbb{N}^+$ , it holds that*

$$687 \quad \text{CPwL}(\mathbb{R}, n) \subseteq \mathcal{NN}(\#input = 1; \text{widthvec} = [n + 1]; \#output = 1). \quad (4.1)$$

688 Lemma 4.1 is a part of Theorem 3.2 in [44] or Lemma 2.2 in [32]. Lemma 4.1 is  
689 Theorem 3.1 in [44] or Lemma 3.4 in [32]. It remains to prove Lemma 4.3.

690 *Proof of Lemma 4.3.* We use the mathematics induction to prove Equation (4.1). First,  
691 consider the case  $n = 1$ . Given any  $f \in \text{CPwL}(\mathbb{R}, 1)$ , there exist  $a_1, a_2, x_0 \in \mathbb{R}$  such that

$$692 \quad f(x) = \begin{cases} a_1(x - x_0) + f(x_0), & \text{if } x \geq x_0, \\ a_2(x_0 - x) + f(x_0), & \text{if } x < x_0. \end{cases}$$

693 Thus,  $f(x) = a_1\sigma(x - x_0) + a_2\sigma(x_0 - x) + f(x_0)$  for any  $x \in \mathbb{R}$ , implying

$$694 \quad f \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2]; \#output = 1).$$

695 Thus, Equation (4.1) holds for  $n = 1$ .

696 Now assume Equation (4.1) holds for  $n = k \in \mathbb{N}^+$ , we would like to show it is also  
697 true for  $n = k + 1$ . Given any  $f \in \text{CPwL}(\mathbb{R}, k + 1)$ , we may assume the biggest breakpoint  
698 of  $f$  is  $x_0$  since it is trivial for the case that  $f$  has no breakpoint. Denote the slopes of  
699 the linear pieces left and right next to  $x_0$  by  $a_1$  and  $a_2$ , respectively. Define

$$700 \quad \tilde{f}(x) := f(x) - (a_2 - a_1)\sigma(x - x_0), \quad \text{for any } x \in \mathbb{R}.$$

701 Then  $\tilde{f}$  has at most  $k$  breakpoints. By the induction hypothesis, we have

$$702 \quad \tilde{f} \in \text{CPwL}(\mathbb{R}, k) \subseteq \mathcal{NN}(\#input = 1; \text{widthvec} = [k + 1]; \#output = 1).$$

703 Thus, there exist  $w_{0,j}, b_{0,j}, w_{1,j}, b_1$  for  $j = 1, 2, \dots, k + 1$  such that

$$704 \quad \tilde{f}(x) = \sum_{j=1}^{k+1} w_{1,j}\sigma(w_{0,j}x + b_{0,j}) + b_1, \quad \text{for any } x \in \mathbb{R}.$$

705 Therefore, for any  $x \in \mathbb{R}$ , we have

$$706 \quad f(x) = (a_2 - a_1)\sigma(x - x_0) + \tilde{f}(x) = (a_2 - a_1)\sigma(x - x_0) + \sum_{j=1}^{k+1} w_{1,j}\sigma(w_{0,j}x + b_{0,j}) + b_1,$$

707 implying  $f \in \mathcal{NN}(\#input = 1; \text{widthvec} = [k + 2]; \#output = 1)$ . Thus, Equation (4.1)  
708 holds for  $k + 1$ , which means we finish the induction process. So we complete the proof.  $\square$

## 4.2 Proof of Proposition 3.1

Now, let us present the detailed proof of Proposition 3.1. Denote  $K = \widetilde{M} \cdot \widetilde{L}$ , where  $\widetilde{M} = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor$ ,  $n = \lfloor \log_3(N+2) \rfloor$ , and  $\widetilde{L} = \lfloor L^{1/d} \rfloor \lfloor n^{1/d} \rfloor$ . Consider the sample set

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

Its size is

$$2\widetilde{M} + 1 = 2\lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor + 1 = \lfloor N^{1/d} \rfloor \cdot \left( (2\lfloor N^{1/d} \rfloor \lfloor L^{1/d} \rfloor - 1) + 1 \right) + 1.$$

By Lemma 4.1 (set  $N_1 = \lfloor N^{1/d} \rfloor$  and  $N_2 = 2\lfloor N^{1/d} \rfloor \lfloor L^{1/d} \rfloor - 1$  therein), there exists

$$\phi_1 \in \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 2(2\lfloor N^{1/d} \rfloor \lfloor L^{1/d} \rfloor - 1) + 1]) \\ = \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{1/d} \rfloor - 1])$$

such that

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

Then, for  $m = 0, 1, \dots, \widetilde{M} - 1$ , we have

$$\phi_1(x) = m, \quad \text{for any } x \in \left[ \frac{m}{\widetilde{M}}, \frac{m+1}{\widetilde{M}} - \delta \cdot \mathbb{1}_{\{m \leq \widetilde{M}-2\}} \right]. \quad (4.2)$$

Now consider another sample set

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

Its size is

$$2\widetilde{L} + 1 = 2\lfloor L^{1/d} \rfloor \lfloor n^{1/d} \rfloor + 1 = \lfloor n^{1/d} \rfloor \cdot \left( (2\lfloor L^{1/d} \rfloor - 1) + 1 \right) + 1.$$

By Lemma 4.1 (set  $N_1 = \lfloor n^{1/d} \rfloor$  and  $N_2 = 2\lfloor L^{1/d} \rfloor - 1$  therein), there exists

$$\phi_2 \in \mathcal{NN}(\text{widthvec} = [2\lfloor n^{1/d} \rfloor, 2(2\lfloor L^{1/d} \rfloor - 1) + 1]) \\ = \mathcal{NN}(\text{widthvec} = [2\lfloor n^{1/d} \rfloor, 4\lfloor L^{1/d} \rfloor - 1])$$

such that

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

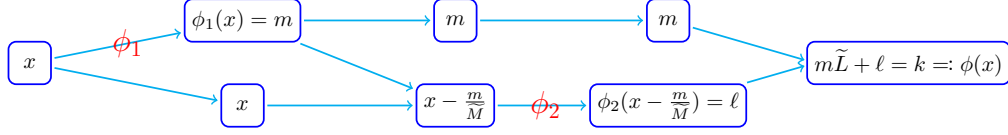


Figure 9: An illustration of the network architecture implementing  $\phi$  based on Equation (4.2) and (4.3) for  $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \leq K-2\}}] = [\frac{m\tilde{L}+\ell}{\tilde{M}\tilde{L}}, \frac{m\tilde{L}+\ell+1}{\tilde{M}\tilde{L}} - \delta \cdot \mathbb{1}_{\{m \leq \tilde{M}-2 \text{ or } \ell \leq \tilde{L}-2\}}]$ , where  $k = m\tilde{L} + \ell$  for  $m = 0, 1, \dots, \tilde{M} - 1$  and  $\ell = 0, 1, \dots, \tilde{L} - 1$ .

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

$$\phi_2(x - \frac{m}{\tilde{M}}) = \ell, \quad \text{for any } x \in [\frac{m\tilde{L}+\ell}{\tilde{M}\tilde{L}}, \frac{m\tilde{L}+\ell+1}{\tilde{M}\tilde{L}} - \delta \cdot \mathbb{1}_{\{\ell \leq \tilde{L}-2\}}]. \quad (4.3)$$

$K = \tilde{M} \cdot \tilde{L}$  implies any  $k \in \{0, 1, \dots, K-1\}$  can be unique represented by  $k = m\tilde{L} + \ell$  for  $m = 0, 1, \dots, \tilde{M} - 1$  and  $\ell = 0, 1, \dots, \tilde{L} - 1$ . Then the desired function  $\phi$  can be implemented by a ReLU network shown in Figure 9.

Clearly,

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

By Lemma 4.2, we have

$$\begin{aligned} \phi_1 &\in \mathcal{NN}(\#input = 1; \text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{1/d} \rfloor - 1]; \#output = 1) \\ &\subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 8\lfloor N^{1/d} \rfloor + 2; \text{depth} \leq \lfloor L^{1/d} \rfloor + 1; \#output = 1) \end{aligned}$$

and

$$\begin{aligned} \phi_2 &\in \mathcal{NN}(\#input = 1; \text{widthvec} = [2\lfloor n^{1/d} \rfloor, 4\lfloor L^{1/d} \rfloor - 1]; \#output = 1) \\ &\subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 8\lfloor n^{1/d} \rfloor + 2; \text{depth} \leq \lfloor L^{1/d} \rfloor + 1; \#output = 1). \end{aligned}$$

Recall that  $n = \lfloor \log_3(N+2) \rfloor \leq N$ . It follows from Figure 9 that  $\phi$  can be implemented by a ReLU network with width

$$\max \{8\lfloor N^{1/d} \rfloor + 2 + 1, 8\lfloor n^{1/d} \rfloor + 2 + 1\} = 8\lfloor N^{1/d} \rfloor + 3$$

and depth

$$(\lfloor L^{1/d} \rfloor + 1) + 2 + (\lfloor L^{1/d} \rfloor + 1) + 1 = 2\lfloor L^{1/d} \rfloor + 5.$$

So we finish the proof.

### 4.3 Proof of Proposition 3.2

The proof of Proposition 3.2 is based on the bit extraction technique in [3, 13]. To simplify the proof, we first prove Lemma 4.4, 4.5, 4.6, and 4.7, which serve as four important intermediate steps. Next, we will apply Lemma 4.7 to prove Proposition 3.2. In fact, we modify this technique to extract the sum of many bits rather than one bit and this modification can be summarized in Lemma 4.4 and 4.5 below.

754 **Lemma 4.4.** For any  $n \in \mathbb{N}^+$ , there exists a function  $\phi$  in

755  $\mathcal{NN}(\#\text{input} = 2; \text{width} \leq (n+1)2^{n+1}; \text{depth} \leq 3; \#\text{output} = 1)$

756 such that: Given any  $\theta_j \in \{0, 1\}$  for  $j = 1, 2, \dots, n$ , we have

757 
$$\phi(\text{bin } 0.\theta_1\theta_2\cdots\theta_n, i) = \sum_{j=1}^i \theta_j, \quad \text{for any } i \in \{0, 1, 2, \dots, n\}. \textcircled{2}$$

758 *Proof.* Define  $\theta = \text{bin } 0.\theta_1\theta_2\cdots\theta_n$ . Clearly,

759 
$$\theta_j = \lfloor 2^j \theta \rfloor / 2 - \lfloor 2^{j-1} \theta \rfloor, \quad \text{for any } j \in \{1, 2, \dots, n\}.$$

760 We shall use a ReLU network to replace  $\lfloor \cdot \rfloor$ . Let  $g \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2)$  be the function  
761 satisfying two conditions:

- 762 •  $g$  matches set of samples

763 
$$\bigcup_{k=0}^{2^n-1} \{(k, k), (k+1-\delta, k)\}, \quad \text{where } \delta = 2^{-(n+1)};$$

- 764 • The breakpoint set of  $g$  is

765 
$$\left( \bigcup_{k=0}^{2^n-1} \{k, k+1-\delta\} \right) \setminus (\{0\} \cup \{2^n - \delta\}).$$

766 Then  $g(x) = \lfloor x \rfloor$  for any  $x \in \bigcup_{k=0}^{2^n-1} [k, k+1-\delta]$ . Clearly,  $\theta = \text{bin } 0.\theta_1\theta_2\cdots\theta_n$  implies

767 
$$2^j \theta \in \bigcup_{k=0}^{2^n-1} [k, k+1-\delta], \quad \text{for any } j \in \{0, 1, 2, \dots, n\}.$$

768 Thus,

769 
$$\theta_j = \lfloor 2^j \theta \rfloor / 2 - \lfloor 2^{j-1} \theta \rfloor = g(2^j \theta) / 2 - g(2^{j-1} \theta), \quad \text{for any } j \in \{1, 2, \dots, n\}. \quad (4.4)$$

770 It is easy to design a ReLU network to output  $\theta_1, \theta_2, \dots, \theta_n$  by Equation (4.4) when  
771 using  $\theta = \text{bin } 0.\theta_1\theta_2\cdots\theta_n$  as the input. However, it is highly non-trivial to construct  
772 a ReLU network to output  $\sum_{j=1}^i \theta_j$  with another input  $i$ , since many operations like  
773 multiplication and comparison are not allowed in designing ReLU networks. Now let us  
774 establish a formula to represent  $\sum_{j=1}^i \theta_j$  in a form of a ReLU network as follows.

775 Define  $\mathcal{T}(n) := \sigma(n+1) - \sigma(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$  for any integer  $n$ . Then, by Equation (4.4)  
776 and the fact  $x_1 x_2 = \sigma(x_1 + x_2 - 1)$  for any  $x_1, x_2 \in \{0, 1\}$ , we have, for  $i = 0, 1, 2, \dots, n$ ,

777 
$$\begin{aligned} \sum_{j=1}^i \theta_j &= \sum_{j=1}^n \theta_j \cdot \mathcal{T}(i-j) = \sum_{j=1}^n \sigma(\theta_j + \mathcal{T}(i-j) - 1) \\ &= \sum_{j=1}^n \sigma(\theta_j + \sigma(i-j+1) - \sigma(i-j) - 1) \\ &= \sum_{j=1}^n \sigma(g(2^j \theta) / 2 - g(2^{j-1} \theta) + \sigma(i-j+1) - \sigma(i-j) - 1). \end{aligned}$$

---

<sup>②</sup>By convention,  $\sum_{j=n}^m a_j = 0$  if  $n > m$ , no matter what  $a_j$  is for each  $j$ .

778 Define

$$779 \quad z_{i,j} := \sigma\left(g(2^j\theta)/2 - g(2^{j-1}\theta) + \sigma(i-j+1) - \sigma(i-j) - 1\right), \quad (4.5)$$

780 for any  $i, j \in \{1, 2, \dots, n\}$ . Then the goal is to design  $\phi$  satisfying

$$781 \quad \phi(\theta, i) = \sum_{j=1}^i \theta_j = \sum_{j=1}^n z_{i,j}, \quad \text{for any } i \in \{0, 1, 2, \dots, n\}. \quad (4.6)$$

782 See Figure 10 for the network architecture implementing the desired function  $\phi$ .

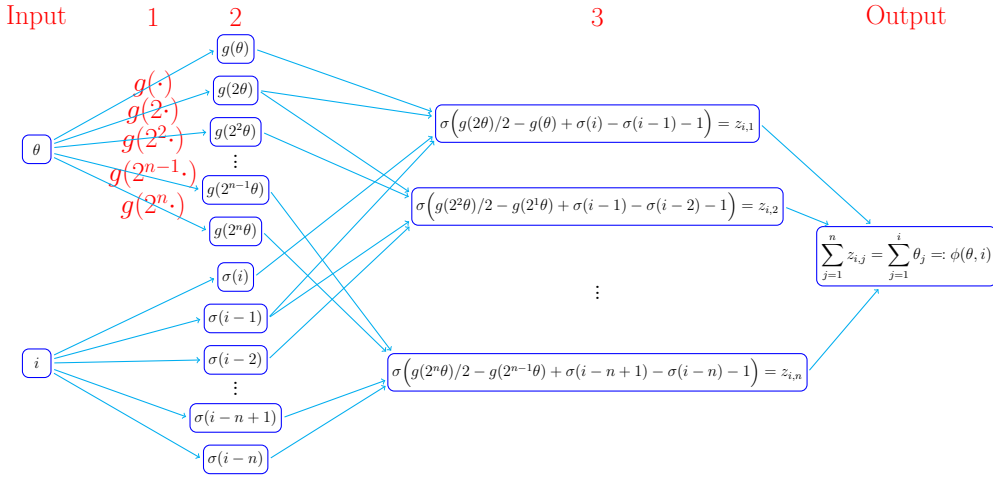


Figure 10: An illustration of the network implementing the desired function  $\phi$  with the input  $[\theta, i]^T = [\text{bin}0.\theta_1\theta_2\cdots\theta_n, i]^T$  for any  $i \in \{0, 1, 2, \dots, n\}$  and  $\theta_1, \theta_2, \dots, \theta_n \in \{0, 1\}$ .  $g(2^j \cdot)$  can be implemented by a one-hidden-layer network with width  $2^{n+1} - 1$  for each  $j \in \{0, 1, 2, \dots, n\}$ . The red numbers above the architecture indicate the order of hidden layers. The network architecture is essentially determined by Equation (4.5) and (4.6), which are valid no matter what  $\theta_1, \theta_2, \dots, \theta_n \in \{0, 1\}$  are. Thus, the desired function  $\phi$  is independent of  $\theta_1, \theta_2, \dots, \theta_n \in \{0, 1\}$ . We omit ReLU ( $\sigma$ ) for a neuron if its output is non-negative without ReLU. Such a simplification are applied to similar figures in this paper.

783 By Lemma 4.3, we have

$$784 \quad g \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2) \subseteq \mathcal{NN}(\#input = 1; \text{widthvec} = [2^{n+1} - 1]; \#output = 1),$$

785 implying

$$786 \quad g(2^j \cdot) \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2) \subseteq \mathcal{NN}(\#input = 1; \text{widthvec} = [2^{n+1} - 1]; \#output = 1),$$

787 for any  $j = 0, 1, 2, \dots, n$ . Clearly, the network in Figure 10 has width

$$788 \quad (n+1)(2^{n+1} - 1) + (n+1) = (n+1)2^{n+1}$$

789 and depth 3. So we finish the proof. □

790 **Lemma 4.5.** For any  $n, L \in \mathbb{N}^+$ , there exists a function  $\phi$  in

791  $\mathcal{NN}(\#input = 2; \text{ width } \leq (n+3)2^{n+1} + 4; \text{ depth } \leq 4L + 2; \#output = 1)$

792 such that: Given any  $\theta_j \in \{0, 1\}$  for  $j = 1, 2, \dots, Ln$ , we have

793 
$$\phi(\text{bin}0.\theta_1\theta_2\cdots\theta_{Ln}, k) = \sum_{j=1}^k \theta_j, \quad \text{for any } k \in \{1, 2, \dots, Ln\}.$$

794 *Proof.* Let  $g_1 \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2)$  be the function satisfying:

- 795 •  $g_1$  matches the set of samples

796 
$$\bigcup_{i=0}^{2^n-1} \{(i, i), (i+1-\delta, i)\}, \quad \text{where } \delta = 2^{-(Ln+1)}.$$

- 797 • The breakpoint set of  $g_1$  is

798 
$$\left( \bigcup_{i=0}^{2^n-1} \{(i, i), (i+1-\delta, i)\} \right) \setminus (\{0\} \cup \{2^n - \delta\}).$$

799 Then  $g_1(x) = \lfloor x \rfloor$  for any  $x \in \bigcup_{i=0}^{2^n-1} [i, i+1-\delta]$ . Note that

800 
$$2^n \cdot \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln} \in \bigcup_{i=0}^{2^n-1} [i, i+1-\delta], \quad \text{for any } \ell \in \{0, 1, \dots, L-1\}.$$

801 Thus, for any  $\ell \in \{0, 1, \dots, L-1\}$ , we have

802 
$$\text{bin}0.\theta_{\ell n+1}\cdots\theta_{\ell n+n} = \frac{\lfloor 2^n \cdot \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln} \rfloor}{2^n} = \frac{g_1(2^n \cdot \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln})}{2^n}. \quad (4.7)$$

803 Define  $g_2(x) := 2^n x - g_1(2^n x)$  for any  $x \in \mathbb{R}$ . Then  $g_2 \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2)$  and

804 
$$\begin{aligned} \text{bin}0.\theta_{(\ell+1)n+1}\cdots\theta_{Ln} &= 2^n \left( \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln} - \text{bin}0.\theta_{\ell n+1}\cdots\theta_{\ell n+n} \right) \\ &= 2^n \left( \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln} - \frac{g_1(2^n \cdot \text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln})}{2^n} \right) = g_2(\text{bin}0.\theta_{\ell n+1}\cdots\theta_{Ln}). \end{aligned} \quad (4.8)$$

805 By Lemma 4.4, there exists

806 
$$\phi_1 \in \mathcal{NN}(\#input = 2; \text{ width } \leq (n+1)2^{n+1}; \text{ depth } \leq 3; \#output = 1)$$

807 such that: For any  $\xi_1, \xi_2, \dots, \xi_n \in \{0, 1\}$ , we have

808 
$$\phi_1(\text{bin}0.\xi_1\xi_2\cdots\xi_n, i) = \sum_{j=1}^i \xi_j, \quad \text{for } i = 0, 1, 2, \dots, n.$$

809 It follows that

810 
$$\phi_1(\text{bin}0.\theta_{\ell n+1}\theta_{\ell n+2}\cdots\theta_{\ell n+n}, i) = \sum_{j=1}^i \theta_{\ell n+j}, \quad \text{for } \ell = 0, 1, \dots, L-1 \text{ and } i = 0, 1, \dots, n. \quad (4.9)$$



811 Define  $\phi_{2,\ell}(x) := \min\{\sigma(x - \ell n), n\}$  for any  $x \in \mathbb{R}$  and  $\ell \in \{0, 1, \dots, L-1\}$ . For any  
 812  $k \in \{1, 2, \dots, Ln\}$ , there exists  $k_1 \in \{0, 1, \dots, L-1\}$  and  $k_2 \in \{1, 2, \dots, n\}$  such that  $k = k_1 n + k_2$ ,  
 813 implying

$$\begin{aligned} \sum_{i=1}^k \theta_i &= \sum_{i=1}^{k_1 n + k_2} \theta_i = \sum_{\ell=0}^{k_1-1} \left( \sum_{j=1}^n \theta_{\ell n + j} \right) + \sum_{\ell=k_1}^{k_1-1} \left( \sum_{j=1}^{k_2} \theta_{\ell n + j} \right) + \sum_{\ell=k_1+1}^{L-1} \left( \sum_{j=1}^0 \theta_{\ell n + j} \right) \\ &= \sum_{\ell=0}^{L-1} \left( \sum_{j=1}^{\min\{\sigma(k - \ell n), n\}} \theta_{\ell n + j} \right) = \sum_{\ell=0}^{L-1} \left( \sum_{j=1}^{\phi_{2,\ell}(k)} \theta_{\ell n + j} \right). \end{aligned} \quad (4.10)$$

815 Then, the desired function  $\phi$  can be implemented by the network architecture in Fig-  
 816 ure 11.

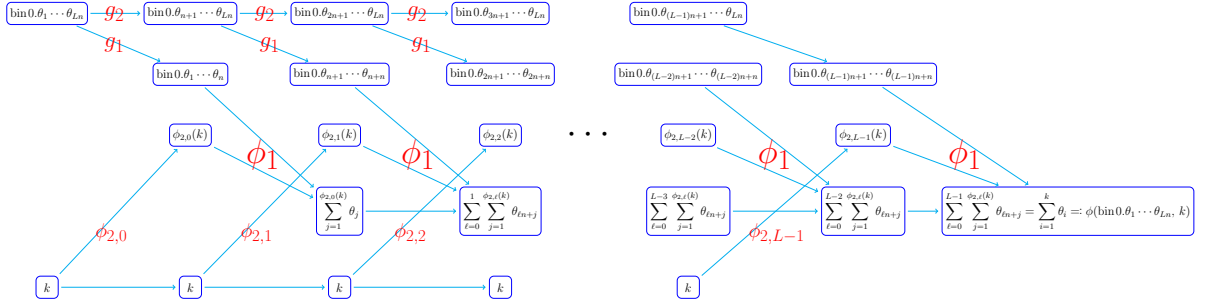


Figure 11: An illustration of the network implementing the desired function  $\phi$  with the input  $[\text{bin } 0. \theta_1 \theta_2 \dots \theta_{Ln}, k]^T$  for any  $k \in \{1, 2, \dots, Ln\}$  and  $\theta_1, \theta_2, \dots, \theta_{Ln} \in \{0, 1\}$ . The network architecture is essentially determined by Equation (4.7), (4.8), (4.9), and (4.10), which are valid no matter what  $\theta_1, \theta_2, \dots, \theta_{Ln} \in \{0, 1\}$  are. Thus, the desired function  $\phi$  is independent of  $\theta_1, \theta_2, \dots, \theta_{Ln} \in \{0, 1\}$ . We omit ReLU ( $\sigma$ ) for a neuron if its output is non-negative without ReLU.

817 By Lemma 4.3, we have

$$818 \quad g_1, g_2 \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2) \subseteq \mathcal{NN}(\# \text{input} = 1; \text{widthvec} = [2^{n+1} - 1]; \# \text{output} = 1).$$

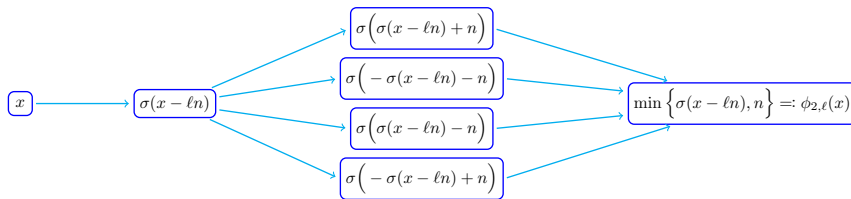


Figure 12: An illustration of the network implementing the desired function  $\phi_{2,\ell}$  for each  $\ell \in \{0, 1, \dots, L-1\}$ , based on  $\min\{y, n\} = \frac{1}{2}(\sigma(y + n) - \sigma(-y - n) - \sigma(y - n) - \sigma(-y + n))$ .

819 Recall that  $\phi_1 \in \mathcal{NN}(\text{width} \leq (n+1)2^{n+1}; \text{depth} \leq 3)$ . As shown in Figure 12,  
 820  $\phi_{2,\ell}(x) \in \mathcal{NN}(\text{width} \leq 4; \text{depth} \leq 2)$  for  $\ell = 0, 1, \dots, L-1$ . Therefore, the network in  
 821 Figure 11 has width

$$822 \quad (2^{n+1} - 1) + (2^{n+1} - 1) + (n+1)2^{n+1} + 1 + 4 + 1 = (n+3)2^{n+1} + 4$$

823 and depth

$$824 \quad 2 + L(1 + 3) = 4L + 2.$$

825 So we finish the proof. □

826 Next, we introduce Lemma 4.6 to map indices to the partial sum of given bits.

827 **Lemma 4.6.** *Given any  $N, L \in \mathbb{N}^+$  and arbitrary  $\theta_{m,k} \in \{0, 1\}$  for  $m = 0, 1, \dots, M - 1$  and*  
 828  *$k = 0, 1, \dots, Ln - 1$ , where  $M = N^2L$  and  $n = \lfloor \log_3(N + 2) \rfloor$ , there exists*

$$829 \quad \phi \in \mathcal{NN}(\#input = 2; \text{ width } \leq 6N + 14; \text{ depth } \leq 5L + 4; \#output = 1)$$

830 such that

$$831 \quad \phi(m, k) = \sum_{j=0}^k \theta_{m,j}, \quad \text{for } m = 0, 1, \dots, M - 1 \text{ and } k = 0, 1, \dots, Ln - 1.$$

832 *Proof.* Define

$$833 \quad y_m := \text{bin}0.\theta_{m,0}\theta_{m,1}\dots\theta_{m,Ln-1}, \quad \text{for } m = 0, 1, \dots, M - 1.$$

834 Consider the sample set  $\{(m, y_m) : m = 0, 1, \dots, M\}$ , whose cardinality is

$$835 \quad M + 1 = N((NL - 1) + 1) + 1.$$

836 By Lemma 4.1 (set  $N_1 = N$  and  $N_2 = NL - 1$  therein), there exists

$$837 \quad \begin{aligned} \phi_1 &\in \mathcal{NN}(\#input = 1; \text{ widthvec } = [2N, 2(NL - 1) + 1]; \#output = 1) \\ &= \mathcal{NN}(\#input = 1; \text{ widthvec } = [2N, 2NL - 1]; \#output = 1) \end{aligned}$$

838 such that

$$839 \quad \phi_1(m) = y_m, \quad \text{for } m = 0, 1, \dots, M - 1.$$

840 By Lemma 4.5, there exists

$$841 \quad \phi_2 \in \mathcal{NN}(\#input = 2; \text{ width } \leq (n + 3)2^{n+1} + 4; \text{ depth } \leq 4L + 2; \#output = 1)$$

842 such that, for any  $\xi_1, \xi_2, \dots, \xi_{Ln} \in \{0, 1\}$ , we have

$$843 \quad \phi_2(\text{bin}0.\xi_1\xi_2\dots\xi_{Ln}, k) = \sum_{j=1}^k \xi_j, \quad \text{for } k = 1, 2, \dots, Ln.$$

844 It follows that, for any  $\xi_0, \xi_1, \dots, \xi_{Ln-1} \in \{0, 1\}$ , we have

$$845 \quad \phi_2(\text{bin}0.\xi_0\xi_1\dots\xi_{Ln-1}, k + 1) = \sum_{j=0}^k \xi_j, \quad \text{for } k = 0, 1, \dots, Ln - 1.$$

846 Thus, for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, Ln - 1$ , we have

$$847 \quad \phi_2(\phi_1(m), k + 1) = \phi_2(y_m, k + 1) = \phi_2(0.\theta_{m,0}\theta_{m,1}\dots\theta_{m,Ln-1}, k + 1) = \sum_{j=0}^k \theta_{m,j}.$$

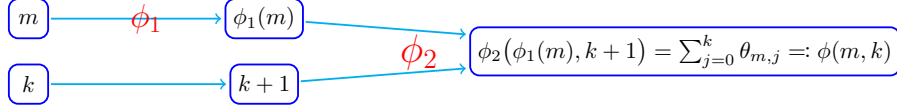


Figure 13: An illustration of the network implementing the desired function  $\phi$  for  $m = 0, 1, \dots, M-1$  and  $k = 0, 1, \dots, Ln-1$ .

Hence, the desired function  $\phi$  can be implemented by the network shown in Figure 13. By Lemma 4.2,  $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL-1]) \subseteq \mathcal{NN}(\text{width} \leq 4N+2; \text{depth} \leq L+1)$ . It holds that

$$(n+3)2^{n+1} + 4 \leq 6 \cdot (3^n) + 2 = 6 \cdot (3^{\lfloor \log_3(N+2) \rfloor}) + 2 \leq 6(N+2) + 2 = 6N + 14,$$

implying

$$\begin{aligned} \phi_2 &\in \mathcal{NN}(\#input = 2; \text{width} \leq (n+3)2^{n+1} + 4; \text{depth} \leq 4L+2; \#output = 1) \\ &\subseteq \mathcal{NN}(\#input = 2; \text{width} \leq 6N + 14; \text{depth} \leq 4L+2; \#output = 1). \end{aligned}$$

Therefore, the network in Figure 13 is with width  $\max\{(4N+2)+1, 6N+14\} = 6N+14$  and depth  $(4L+2)+1+(L+1) = 5L+4$ . So we finish the proof.  $\square$

Next, we apply Lemma 4.6 to prove Lemma 4.7 below, which is a key intermediate conclusion to prove Proposition 3.2.

**Lemma 4.7.** For any  $\varepsilon > 0$  and  $N, L \in \mathbb{N}^+$ , denote  $M = N^2L$  and  $n = \lfloor \log_3(N+2) \rfloor$ . Assume  $y_{m,k} \geq 0$  for  $m = 0, 1, \dots, M-1$  and  $k = 0, 1, \dots, Ln-1$  are samples with

$$|y_{m,k} - y_{m,k-1}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M-1 \quad \text{and} \quad k = 1, 2, \dots, Ln-1.$$

Then there exists  $\phi \in \mathcal{NN}(\#input = 2; \text{width} \leq 16N+30; \text{depth} \leq 5L+7; \#output = 1)$  such that

- (i)  $|\phi(m, k) - y_{m,k}| \leq \varepsilon$  for  $m = 0, 1, \dots, M-1$  and  $k = 0, 1, \dots, Ln-1$ ;
- (ii)  $0 \leq \phi(x_1, x_2) \leq \max\{y_{m,k} : m = 0, 1, \dots, M-1 \text{ and } k = 0, 1, \dots, Ln-1\}$  for any  $x_1, x_2 \in \mathbb{R}$ .

*Proof.* Define

$$a_{m,k} := \lfloor y_{m,k}/\varepsilon \rfloor, \quad \text{for } m = 0, 1, \dots, M-1 \quad \text{and} \quad k = 0, 1, \dots, Ln-1.$$

We will construct a function implemented by a ReLU network to map the index  $(m, k)$  to  $a_{m,k}\varepsilon$  for  $m = 0, 1, \dots, M-1$  and  $k = 0, 1, \dots, Ln-1$ .

Define  $b_{m,0} := 0$  and  $b_{m,k} := a_{m,k} - a_{m,k-1}$  for  $m = 0, 1, \dots, M-1$  and  $k = 1, 2, \dots, Ln-1$ . Since  $|y_{m,k} - y_{m,k-1}| \leq \varepsilon$  for all  $m$  and  $k$ , we have  $b_{m,k} \in \{-1, 0, 1\}$ . Hence, there exist  $c_{m,k} \in \{0, 1\}$  and  $d_{m,k} \in \{0, 1\}$  such that  $b_{m,k} = c_{m,k} - d_{m,k}$ , which implies

$$\begin{aligned} a_{m,k} &= a_{m,0} + \sum_{i=1}^k (a_{m,i} - a_{m,i-1}) = a_{m,0} + \sum_{i=1}^k b_{m,i} = a_{m,0} + \sum_{i=0}^k b_{m,i} \\ &= a_{m,0} + \sum_{i=0}^k c_{m,i} - \sum_{i=0}^k d_{m,i}, \end{aligned}$$

874 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, Ln - 1$ .

875 Consider the sample set

$$876 \quad \{(m, a_{m,0}) : m = 0, 1, \dots, M - 1\} \cup \{(M, 0)\}.$$

877 Its size is  $M + 1 = N \cdot ((NL - 1) + 1) + 1$ , by Lemma 4.1 (set  $N_1 = N$  and  $N_2 = NL - 1$   
878 therein), there exists

$$879 \quad \psi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(NL - 1) + 1]) = \mathcal{NN}(\text{widthvec} = [2N, 2NL - 1])$$

880 such that

$$881 \quad \psi_1(m) = a_{m,0}, \quad \text{for } m = 0, 1, \dots, M - 1.$$

882 By Lemma 4.6, there exist  $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 6N + 14; \text{depth} \leq 5L + 4)$  such that

$$883 \quad \psi_2(m, k) = \sum_{i=0}^k c_{m,i} \quad \text{and} \quad \psi_3(m, k) = \sum_{i=0}^k d_{m,i},$$

884 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, Ln - 1$ . Hence, it holds that

$$885 \quad a_{m,k} = a_{m,0} + \sum_{i=0}^k c_{m,i} - \sum_{i=0}^k d_{m,i} = \psi_1(m) + \psi_2(m, k) - \psi_3(m, k), \quad (4.11)$$

886 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, Ln - 1$ .

887 Define

$$888 \quad y_{\max} := \max\{y_{m,k} : m = 0, 1, \dots, M - 1 \quad \text{and} \quad k = 0, 1, \dots, Ln - 1\}.$$

889 Then the desired function can be implemented by two sub-networks shown in Figure 14.

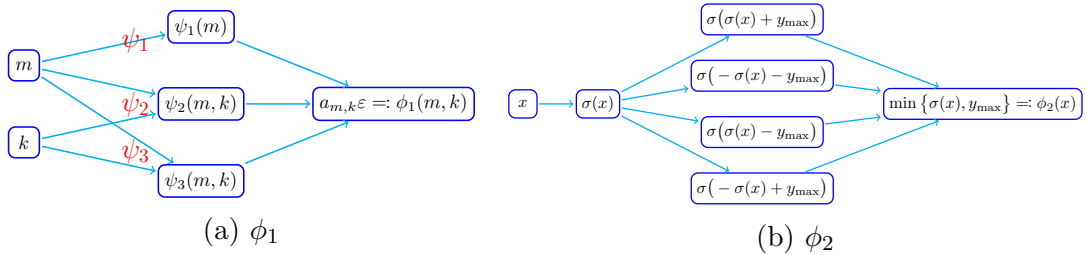


Figure 14: Illustrations of two sub-networks implementing the desired function  $\phi = \phi_2 \circ \phi_1$  for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, Ln - 1$ , based on Equation (4.11) and the fact  $\min\{x_1, x_2\} = \frac{x_1 + x_2 - |x_1 - x_2|}{2} = \frac{\sigma(x_1 + x_2) - \sigma(-x_1 - x_2) - \sigma(x_1 - x_2) - \sigma(-x_1 + x_2)}{2}$ .

890 By Lemma 4.2,

$$891 \quad \begin{aligned} & \psi_1 \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 2NL - 1]; \#output = 1) \\ & \subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 4N + 2; \text{depth} \leq L + 1; \#output = 1). \end{aligned}$$

892 Recall that  $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 6N + 14; \text{depth} \leq 5L + 4)$ . Thus,  $\phi_1 \in \mathcal{NN}(\text{width} \leq$   
893  $(4N + 2) + 2(6N + 14) = 16N + 30; \text{depth} \leq (5L + 4) + 1 = 5L + 5)$  as shown in Figure 14.

894 And it is clear that  $\phi_2 \in \mathcal{NN}(\text{width} \leq 4; \text{depth} \leq 2)$ , implying  $\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}(\text{width} \leq$   
 895  $16N + 30; \text{depth} \leq (5L + 5) + 2 = 5L + 7)$ .

896 Clearly,  $0 \leq \phi(x_1, x_2) \leq y_{\max}$  for any  $x_1, x_2 \in \mathbb{R}$ , since  $\phi(x_1, x_2) = \phi_2 \circ \phi_1(x_1, x_2) =$   
 897  $\max\{\sigma(\phi_1(x_1, x_2)), y_{\max}\}$ .

898 Note that  $0 \leq a_{m,k}\varepsilon = \lfloor y_{m,k}/\varepsilon \rfloor \varepsilon \leq y_{\max}$ . Then we have  $\phi(m, k) = \phi_2 \circ \phi_1(m, k) =$   
 899  $\phi_2(a_{m,k}\varepsilon) = \max\{\sigma(a_{m,k}\varepsilon), y_{\max}\} = a_{m,k}\varepsilon$ . Therefore,

$$900 \quad |\phi(m, k) - y_{m,k}| = |a_{m,k}\varepsilon - y_{m,k}| = |\lfloor y_{m,k}/\varepsilon \rfloor \varepsilon - y_{m,k}| \leq \varepsilon,$$

901 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, \widehat{L} - 1$ . Hence, we finish the proof.  $\square$

902 Finally, we apply Lemma 4.7 to prove Proposition 3.2.

903 *Proof of Proposition 3.2.* Denote  $M = N^2L$ ,  $n = \lfloor \log_3(N + 2) \rfloor$ , and  $\widehat{L} = Ln$ . We may  
 904 assume  $J = M\widehat{L}n = M\widehat{L}$  since we can set  $y_{J-1} = y_J = y_{J+1} = \dots = y_{M\widehat{L}-1}$  if  $J < M\widehat{L}$ .

905 Consider the sample set

$$906 \quad \{(m\widehat{L}, m) : m = 0, 1, \dots, M\} \cup \{(m\widehat{L} + \widehat{L} - 1, m) : m = 0, 1, \dots, M - 1\}.$$

907 Its size is  $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$ . By Lemma 4.1 (set  $N_1 = N$  and  $N_2 = NL - 1$   
 908 therein), there exist

$$909 \quad \phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL - 1) + 1]) = \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1])$$

910 such that

- 911 •  $\phi_1(M\widehat{L}) = M$  and  $\phi_1(m\widehat{L}) = \phi_1(m\widehat{L} + \widehat{L} - 1) = m$  for  $m = 0, 1, \dots, M - 1$ .
- 912 •  $\phi_1$  is linear on each interval  $[m\widehat{L}, m\widehat{L} + \widehat{L} - 1]$  for  $m = 0, 1, \dots, M - 1$ .

913 It follows that

$$914 \quad \phi_1(j) = m, \quad \text{and} \quad j - \widehat{L}\phi_1(j) = k, \quad \text{where } j = m\widehat{L} + k, \quad (4.12)$$

915 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, \widehat{L} - 1$ .

916 Note that any number  $j$  in  $\{0, 1, \dots, J - 1\}$  can be uniquely indexed as  $j = m\widehat{L} + k$   
 917 for  $m = 0, 1, \dots, M - 1$  and  $k = 0, 1, \dots, \widehat{L} - 1$ . So we can denote  $y_j = y_{m\widehat{L}+k}$  as  $y_{m,k}$ . Then  
 918 by Lemma 4.7, there exists  $\phi_2 \in \mathcal{NN}(\text{width} \leq 16N + 30; \text{depth} \leq 5L + 7)$  such that

$$919 \quad |\phi_2(m, k) - y_{m,k}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M - 1 \quad \text{and} \quad k = 0, 1, \dots, \widehat{L} - 1, \quad (4.13)$$

920 and

$$921 \quad 0 \leq \phi_2(x_1, x_2) \leq y_{\max}, \quad \text{for any } x_1, x_2 \in \mathbb{R}, \quad (4.14)$$

922 where  $y_{\max} := \max\{y_{m,k} : m = 0, 1, \dots, M - 1 \text{ and } k = 0, 1, \dots, \widehat{L} - 1\} = \max\{y_j : j =$   
 923  $0, 1, \dots, M\widehat{L} - 1\}$ .

924 By Lemma 4.2,

$$925 \quad \begin{aligned} &\phi_1 \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 4NL - 1]; \#output = 1) \\ &\subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 8N + 2; \text{depth} \leq L + 1; \#output = 1). \end{aligned}$$

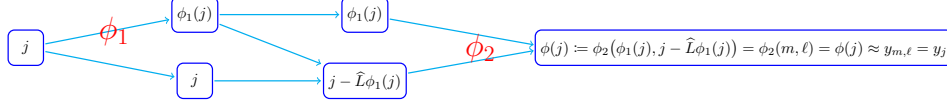


Figure 15: An illustration of the ReLU network implementing the desired function  $\phi$  based Equation (4.12). The index  $j \in \{0, 1, \dots, M\widehat{L}-1\}$  is unique represented by  $j = mL+k$  for  $m = 0, 1, \dots, M-1$  and  $k = 0, 1, \dots, \widehat{L}-1$ .

926 Recall that  $\phi_2 \in \mathcal{NN}(\text{width} \leq 16N + 30; \text{depth} \leq 5L + 7)$ . So  $\phi \in \mathcal{NN}(\text{width} \leq 16N +$   
 927  $30; \text{depth} \leq (L+1) + 2 + (5L+7) = 6L+10)$  as shown in Figure 15.

928 Equation (4.14) implies

$$929 \quad 0 \leq \phi(x) \leq y_{\max}, \quad \text{for any } x \in \mathbb{R},$$

930 since  $\phi$  is given by  $\phi(x) = \phi_2(\phi_1(x), x - \widehat{L}\phi_1(x))$ .

931 Represent  $j \in \{0, 1, \dots, M\widehat{L}-1\}$  via  $j = m\widehat{L} + k$  for  $m = 0, 1, \dots, M-1$  and  $k =$   
 932  $0, 1, \dots, \widehat{L}-1$ . Then, by Equation (4.13), we have

$$933 \quad |\phi(j) - y_j| = |\phi_2(\phi_1(j), j - \widehat{L}\phi_1(j)) - y_j| = |\phi_2(m, k) - y_{m,k}| \leq \varepsilon,$$

934 for any  $j \in \{0, 1, \dots, M\widehat{L}-1\} = \{0, 1, \dots, J-1\}$ . So we finish the proof.  $\square$

935 We would like to remark that the key idea in the proof of Proposition 3.2 is the bit  
 936 extraction technique in Lemma 4.5, which allows us to store  $Ln$  bits in a binary number  
 937  $\text{bin}0.\theta_1\theta_2\cdots\theta_{Ln}$  and extract each bit  $\theta_i$ . The extraction operator can be efficiently carried  
 938 out via a deep ReLU neural network demonstrating the power of depth.

## 939 5 Conclusion and future work

940 This paper aims at a quantitative and optimal approximation rate for ReLU net-  
 941 works in terms of the width and depth to approximate continuous functions. It is  
 942 shown by construction that ReLU networks with width  $\mathcal{O}(N)$  and depth  $\mathcal{O}(L)$  can  
 943 approximate an arbitrary continuous function on  $[0, 1]^d$  with an approximation rate  
 944  $\mathcal{O}(\omega_f((N^2L^2 \ln N)^{-1/d}))$ . By connecting the approximation property to VC-dimension,  
 945 we prove that such a rate is optimal for Hölder continuous functions on  $[0, 1]^d$  in terms  
 946 of the width and depth separately, and hence this rate is also optimal for the whole  
 947 continuous function class. We also extend our analysis to general continuous functions  
 948 on any bounded subset of  $\mathbb{R}^d$ . We would like to remark that our analysis was based on  
 949 the fully connected feed-forward neural networks and the ReLU activation function. It  
 950 would be very interesting to extend our conclusions to neural networks with other types  
 951 of architectures (e.g., convolutional neural networks) and activation functions (e.g., tanh  
 952 and sigmoid functions).

## 953 Acknowledgments

954 Z. Shen is supported by Tan Chin Tuan Centennial Professorship. H. Yang was  
 955 partially supported by the US National Science Foundation under award DMS-1945029.

S. Zhang is supported by a Postdoctoral Fellowship under NUS ENDOWMENT FUND (EXP WBS) (01 651).

## References

- [1] M. ANTHONY AND P. L. BARTLETT, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, New York, NY, USA, 1st ed., 2009.
- [2] A. R. BARRON, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.
- [3] P. BARTLETT, V. MAIOROV, AND R. MEIR, *Almost linear VC-dimension bounds for piecewise polynomial networks*, Neural Computation, 10 (1998), pp. 2159–2173.
- [4] M. BIANCHINI AND F. SCARSELLI, *On the complexity of neural network classifiers: A comparison between shallow and deep architectures*, IEEE Transactions on Neural Networks and Learning Systems, 25 (2014), pp. 1553–1565.
- [5] L. CHEN AND C. WU, *A note on the expressive power of deep rectified linear unit networks in high-dimensional spaces*, Mathematical Methods in the Applied Sciences, 42 (2019), pp. 3400–3404.
- [6] G. CYBENKO, *Approximation by superpositions of a sigmoidal function*, MCSS, 2 (1989), pp. 303–314.
- [7] W. E, C. MA, AND Q. WANG, *A Priori Estimates of the Population Risk for Residual Networks*, arXiv e-prints, (2019).
- [8] W. E, C. MA, AND L. WU, *A priori estimates of the population risk for two-layer neural networks*, Communications in Mathematical Sciences, 17 (2019), pp. 1407–1425.
- [9] W. E AND Q. WANG, *Exponential convergence of the deep neural network approximation for analytic functions*, CoRR, abs/1807.00297 (2018).
- [10] W. E AND S. WOJTOWYTSCH, *A priori estimates for classification problems using neural networks*, arXiv e-prints, (2020).
- [11] —, *On the Banach spaces associated with multi-layer ReLU networks: Function representation, approximation theory and gradient descent dynamics*, arXiv e-prints, (2020).
- [12] —, *Representation formulas and pointwise properties for Barron functions*, arXiv e-prints, (2020).
- [13] N. HARVEY, C. LIAW, AND A. MEHRABIAN, *Nearly-tight VC-dimension bounds for piecewise linear neural networks*, in Proceedings of the 2017 Conference on Learning Theory, S. Kale and O. Shamir, eds., vol. 65 of Proceedings of Machine Learning Research, Amsterdam, Netherlands, 07–10 Jul 2017, PMLR, pp. 1064–1068.



- 991 [14] J. HE, X. JIA, J. XU, L. ZHANG, AND L. ZHAO, *Make  $\ell_1$  regularization effective*  
992 *in training sparse CNN*, Computational Optimization and Applications, 77 (2020),  
993 p. 163–182.
- 994 [15] K. HORNIK, *Approximation capabilities of multilayer feedforward networks*, Neural  
995 Networks, 4 (1991), pp. 251–257.
- 996 [16] K. HORNIK, M. STINCHCOMBE, AND H. WHITE, *Multilayer feedforward networks*  
997 *are universal approximators*, Neural Networks, 2 (1989), pp. 359–366.
- 998 [17] K. KAWAGUCHI, *Deep learning without poor local minima*, in Advances in Neu-  
999 ral Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg,  
1000 I. Guyon, and R. Garnett, eds., Curran Associates, Inc., 2016, pp. 586–594.
- 1001 [18] K. KAWAGUCHI AND Y. BENGIO, *Depth with nonlinearity creates no bad local*  
1002 *minima in resnets*, Neural Networks, 118 (2019), pp. 167–174.
- 1003 [19] M. J. KEARNS AND R. E. SCHAPIRE, *Efficient distribution-free learning of prob-*  
1004 *abilistic concepts*, J. Comput. Syst. Sci., 48 (1994), pp. 464–497.
- 1005 [20] Q. LI, T. LIN, AND Z. SHEN, *Deep learning via dynamical systems: An approxi-*  
1006 *mation perspective*, Journal of European Mathematical Society, (to appear).
- 1007 [21] Q. LI, C. TAI, AND W. E, *Stochastic modified equations and dynamics of stochas-*  
1008 *tic gradient algorithms I: Mathematical foundations*, Journal of Machine Learning  
1009 Research, 20 (2019), pp. 1–47.
- 1010 [22] S. LIANG AND R. SRIKANT, *Why deep neural networks?*, CoRR, abs/1610.04161  
1011 (2016).
- 1012 [23] H. LIN AND S. JEGELKA, *Resnet with one-neuron hidden layers is a universal*  
1013 *approximator*, in Advances in Neural Information Processing Systems, S. Bengio,  
1014 H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.,  
1015 vol. 31, Curran Associates, Inc., 2018.
- 1016 [24] J. LU, Z. SHEN, H. YANG, AND S. ZHANG, *Deep network approximation for*  
1017 *smooth functions*, arXiv e-prints, (2020).
- 1018 [25] Z. LU, H. PU, F. WANG, Z. HU, AND L. WANG, *The expressive power of neural*  
1019 *networks: A view from the width*, in Advances in Neural Information Processing  
1020 Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-  
1021 wanathan, and R. Garnett, eds., Curran Associates, Inc., 2017, pp. 6231–6239.
- 1022 [26] H. MONTANELLI, H. YANG, AND Q. DU, *Deep ReLU networks overcome the curse*  
1023 *of dimensionality for bandlimited functions*, Journal of Computational Mathematics,  
1024 (to appear).
- 1025 [27] G. F. MONTUFAR, R. PASCANU, K. CHO, AND Y. BENGIO, *On the number of*  
1026 *linear regions of deep neural networks*, in Advances in Neural Information Processing  
1027 Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q.  
1028 Weinberger, eds., Curran Associates, Inc., 2014, pp. 2924–2932.

- 1029 [28] B. NEYSHABUR, Z. LI, S. BHOJANAPALLI, Y. LECUN, AND N. SREBRO, *The*  
1030 *role of over-parametrization in generalization of neural networks*, in International  
1031 Conference on Learning Representations, 2019.
- 1032 [29] Q. N. NGUYEN AND M. HEIN, *The loss surface of deep and wide neural networks*,  
1033 CoRR, abs/1704.08045 (2017).
- 1034 [30] P. PETERSEN AND F. VOIGTLAENDER, *Optimal approximation of piecewise smooth*  
1035 *functions using deep ReLU neural networks*, Neural Networks, 108 (2018), pp. 296–  
1036 330.
- 1037 [31] A. SAKURAI, *Tight bounds for the VC-dimension of piecewise polynomial networks*,  
1038 in Advances in Neural Information Processing Systems, Neural information process-  
1039 ing systems foundation, 1999, pp. 323–329.
- 1040 [32] Z. SHEN, H. YANG, AND S. ZHANG, *Nonlinear approximation via compositions*,  
1041 Neural Networks, 119 (2019), pp. 74–84.
- 1042 [33] —, *Deep network approximation characterized by number of neurons*, Communi-  
1043 cations in Computational Physics, 28 (2020), pp. 1768–1811.
- 1044 [34] Z. SHEN, H. YANG, AND S. ZHANG, *Deep Network With Approximation Error*  
1045 *Being Reciprocal of Width to Power of Square Root of Depth*, Neural Computation,  
1046 33 (2021), pp. 1005–1036.
- 1047 [35] Z. SHEN, H. YANG, AND S. ZHANG, *Neural network approximation: Three hidden*  
1048 *layers are enough*, Neural Networks, (to appear).
- 1049 [36] J. W. SIEGEL AND J. XU, *Approximation rates for neural networks with general*  
1050 *activation functions*, Neural Networks, 128 (2020), pp. 313–321.
- 1051 [37] J. W. SIEGEL AND J. XU, *Optimal Approximation Rates and Metric Entropy of*  
1052 *ReLU<sup>k</sup> and Cosine Networks*, arXiv e-prints, (2021).
- 1053 [38] P. URYSOHN, *über die mächtigkeit der zusammenhängenden mengen*, Mathematis-  
1054 che Annalen, 94 (1925), p. 262–295.
- 1055 [39] H. WHITNEY, *Analytic extensions of differentiable functions defined in closed sets*,  
1056 Transactions of the American Mathematical Society, 36 (1934), pp. 63–89.
- 1057 [40] D. YAROTSKY, *Error bounds for approximations with deep ReLU networks*, Neural  
1058 Networks, 94 (2017), pp. 103–114.
- 1059 [41] D. YAROTSKY, *Optimal approximation of continuous functions by very deep ReLU*  
1060 *networks*, in Proceedings of the 31st Conference On Learning Theory, S. Bubeck,  
1061 V. Perchet, and P. Rigollet, eds., vol. 75 of Proceedings of Machine Learning Re-  
1062 search, PMLR, 06–09 Jul 2018, pp. 639–649.
- 1063 [42] D. YAROTSKY, *Elementary superexpressive activations*, arXiv e-prints, (2021).

- 1064 [43] D. YAROTSKY AND A. ZHEVNERCHUK, *The phase diagram of approximation rates*  
1065 *for deep neural networks*, in Advances in Neural Information Processing Systems,  
1066 H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., vol. 33,  
1067 Curran Associates, Inc., 2020, pp. 13005–13015.
- 1068 [44] S. ZHANG, *Deep neural network approximation via function compositions*, PhD  
1069 Thesis, National University of Singapore, (2020). URL [https://scholarbank.](https://scholarbank.nus.edu.sg/handle/10635/186064)  
1070 [nus.edu.sg/handle/10635/186064](https://scholarbank.nus.edu.sg/handle/10635/186064).