

Deep Network Approximation Characterized by Number of Neurons*

Zuowei Shen[†] Haizhao Yang[‡] Shijun Zhang[§]

Abstract

This paper quantitatively characterizes the approximation power of deep feed-forward neural networks (FNNs) in terms of the number of neurons. It is shown by construction that ReLU FNNs with width $\mathcal{O}(\max\{d\lfloor N^{1/d} \rfloor, N+1\})$ and depth $\mathcal{O}(L)$ can approximate an arbitrary Hölder continuous function of order $\alpha \in (0, 1]$ on $[0, 1]^d$ with a nearly tight approximation rate $\mathcal{O}(\sqrt{d}N^{-2\alpha/d}L^{-2\alpha/d})$ measured in L^p -norm for any $N, L \in \mathbb{N}^+$ and $p \in [1, \infty]$. More generally for an arbitrary continuous function f on $[0, 1]^d$ with a modulus of continuity $\omega_f(\cdot)$, the constructive approximation rate is $\mathcal{O}(\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))$. We also extend our analysis to f on irregular domains or those localized in an ε -neighborhood of a $d_{\mathcal{M}}$ -dimensional smooth manifold $\mathcal{M} \subseteq [0, 1]^d$ with $d_{\mathcal{M}} \ll d$. Especially, in the case of an essentially low-dimensional domain, we show an approximation rate $\mathcal{O}(\omega_f(\frac{\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + \varepsilon) + \sqrt{d}\omega_f(\frac{\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}))$ for ReLU FNNs to approximate f in the ε -neighborhood, where $d_\delta = \mathcal{O}(d_{\mathcal{M}}\frac{\ln(d/\delta)}{\delta^2})$ for any $\delta \in (0, 1)$ as a relative error for a projection to approximate an isometry when projecting \mathcal{M} to a d_δ -dimensional domain.

Key words. Deep ReLU Neural Networks, Hölder Continuity, Modulus of Continuity, Approximation Theory, Low-Dimensional Manifold, Parallel Computing.

1 Introduction

The approximation theory of neural networks has been an active research topic in the past few decades. Previously, as a special kind of ridge function approximation, shallow neural networks with one hidden layer and various activation functions (e.g., wavelets pursuits [10, 45], adaptive splines [19, 54], radial basis functions [8, 18, 25, 52, 64], sigmoid functions [7, 13–15, 29, 37, 38, 41, 44]) were widely discussed and admit good approximation properties, e.g., the universal approximation property [16, 29, 30], lessening the curse

*Submitted to the editors DATE.

[†]Department of Mathematics, National University of Singapore (matzuows@nus.edu.sg).

[‡]Department of Mathematics, Purdue University (haizhao@purdue.edu).

[§]Department of Mathematics, National University of Singapore (zhangshijun@u.nus.edu).

29 of dimensionality [4, 21, 22], and providing attractive approximation rate in nonlinear
 30 approximation [10, 18, 19, 25, 45, 54, 64].

31 The introduction of deep networks with more than one hidden layers has made sig-
 32 nificant impacts in many fields in computer science and engineering including computer
 33 vision [35] and natural language processing [1]. New scientific computing tools based on
 34 deep networks have also emerged and facilitated large-scale and high-dimensional prob-
 35 lems that were impractical previously [20, 24]. The design of deep ReLU FNNs is the key
 36 of such a revolution. These breakthroughs have stimulated broad research topics from
 37 different points of views to study the power of deep ReLU FNNs, e.g. in terms of combi-
 38 natorics [50], topology [6], Vapnik-Chervonenkis (VC) dimension [5, 27, 57], fat-shattering
 39 dimension [2, 34], information theory [53], classical approximation theory [4, 16, 30, 61, 66],
 40 optimization [32, 33, 51] etc.

41 Particularly in approximation theory, **non-quantitative and asymptotic** approx-
 42 imation rates of ReLU FNNs have been proposed for various types of functions. For
 43 example, smooth functions [23, 39, 43, 65], piecewise smooth functions [53], band-limited
 44 functions [49], continuous functions [66], solutions to partial differential equations [31].
 45 However, to the best of our knowledge, existing theories [17, 23, 39, 43, 47, 49, 53, 62, 65, 66]
 46 can only provide implicit formulas in the sense that the approximation error contains
 47 an unknown prefactor, or work only for sufficiently large N and L larger than some
 48 unknown numbers. For example, [66] estimated an approximation rate $c(d)L^{-2\alpha/d}$ via a
 49 narrow and deep ReLU FNN, where $c(d)$ is an unknown number depending on d , and
 50 L is required to be larger than a sufficiently large unknown number \mathcal{L} . For another
 51 example, given an approximation error ε , [53] proved the existence of a ReLU FNN with
 52 a constant but still unknown number of layers approximating a C^β function within the
 53 target error. These works can be divided into two cases: 1) FNNs with varying width
 54 and only one hidden layer [18, 25, 40, 64] (visualized by the region in  in Figure 1); 2)
 55 FNNs with a fixed width of $\mathcal{O}(d)$ and a varying depth larger than an unknown number
 56 \mathcal{L} [43, 66] (represented by the region in  in Figure 1).

57 As far as we know, the first **quantitative and non-asymptotic** approximation
 58 rate of deep ReLU FNNs was obtained in [61]. Specifically, [61] identified an explicit
 59 formulas of the approximation rate

$$60 \quad \begin{cases} 2\lambda N^{-2\alpha}, & \text{when } L \geq 2 \text{ and } d = 1, \\ 2(2\sqrt{d})^\alpha \lambda N^{-2\alpha/d}, & \text{when } L \geq 3 \text{ and } d \geq 2, \end{cases} \quad (1.1)$$

61 for ReLU FNNs with an arbitrary width $N \in \mathbb{N}^+$ and a fixed depth $L \in \mathbb{N}^+$ to approximate
 62 a Hölder continuous function f of order α with a Hölder constant λ (visualized in the
 63 region shown by  in Figure 1). The approximation rate $\mathcal{O}(N^{-2\alpha/d})$ is tight in terms
 64 of N and increasing L cannot improve the approximation rate in N . The success of deep
 65 FNNs in a broad range of applications has motivated a well-known conjecture that the
 66 depth L has an important role in improving the approximation power of deep FNNs.
 67 In particular, a very important question in practice would be, given an arbitrary L
 68 and N , what is the explicit formula to characterize the approximation error so as to see
 69 whether the network is large enough to meet the accuracy requirement. Due to the highly

70 nonlinear structure of deep FNNs, it is still a challenging open problem to characterize
 71 N and L simultaneously in the approximation rate.

72 To answer the question just above, we establish the first framework that is able to
 73 quantify the approximation power of deep ReLU FNNs essentially with arbitrary width
 74 N and depth L , achieving a nearly optimal approximation rate, $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$,
 75 for continuous functions $f \in C([0, 1]^d)$. Our result is based on new analysis techniques
 76 merely based on the structure of FNNs and a modified bit extraction technique inspired
 77 by [5], instead of designing FNNs to approximate traditional approximation basis like
 78 polynomials and splines as in the existing literature [26, 39, 43, 47, 48, 53, 55, 56, 59, 62, 65,
 79 66]. The approximation rate obtained here admits an explicit formula to compute the
 80 prefactor when $\omega_f(\cdot)$ is known. For example, in the case of Hölder continuous functions of
 81 order α with a Hölder constant λ (denoted as the class $B_\lambda(C^\alpha([0, 1]^d))$), $\omega_f(r) \leq \lambda r^\alpha$ for
 82 $r \geq 0$, resulting in the approximation rate $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ as mentioned previously.
 83 As a consequence, existing works for the function class $C([0, 1]^d)$ are special cases of our
 84 result (see Figure 1 for a comparison).

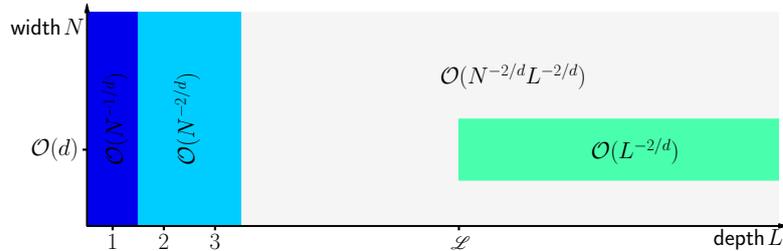


Figure 1: A summary of existing and our new results on the approximation rate of ReLU FNNs for continuous functions. Existing results [18, 25, 40, 43, 61, 64, 66] are applicable in the areas in ■, ■, and ■; our new result is suitable for almost all areas when $L \geq 2$.

85 Our key contributions can be summarized as follows.

- 86 1. Upper bound: We provide a quantitative and non-asymptotic approximation rate
 87 $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$ in terms of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ for functions in
 88 $C([0, 1]^d)$ in Theorem 1.1.
- 89 2. Lower bound: Through the nearly tight VC-dimension bounds of ReLU FNNs [27],
 90 we show that the approximation rate $19\sqrt{d}\omega_f(N^{-2\alpha/d}L^{-2\alpha/d})$ in terms of N and L
 91 is nearly optimal for $B_\lambda(C^\alpha([0, 1]^d))$ in Theorem 2.3.
- 92 3. The approximation rate in terms of the width and depth in this paper is more
 93 generic and useful than the one characterized by the number of nonzero parameters
 94 denoted as W in the literature. First, the characterization in terms of width and
 95 depth implies the one in terms of W , while it is not true the other way around.
 96 Second, our theory can provide practical guidance for choosing network sizes in
 97 realistic applications while theories in terms of W cannot tell how large a network

98 should be to guarantee a target accuracy, since there are too many networks of
 99 different sizes sharing the same number of parameters but with different accuracies.

100 4. Finally, three aspects of neural networks in practice are discussed: 1) neural net-
 101 work approximation in a high-dimensional irregular domain; 2) neural network
 102 approximation in the case of a low-dimensional data structure; 3) the optimal
 103 ReLU FNN in parallel computation.

104 Our main result, Theorem 1.1 below, shows that ReLU FNNs with width $\mathcal{O}(N)$
 105 and depth $\mathcal{O}(L)$ can approximate f with an approximation rate $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$,
 106 where $\omega_f(\cdot)$ is the modulus of continuity of f defined via

$$107 \quad \omega_f(r) := \sup \{ |f(\mathbf{x}) - f(\mathbf{y})| : \mathbf{x}, \mathbf{y} \in [0, 1]^d, \|\mathbf{x} - \mathbf{y}\|_2 \leq r \}, \quad \text{for any } r \geq 0.$$

108 **Theorem 1.1.** *Given $f \in C([0, 1]^d)$, for any $L \in \mathbb{N}^+$, $N \in \mathbb{N}^+$, and $p \in [1, \infty]$, there exists
 109 a function ϕ implemented by a ReLU FNN with width $C_1 \max \{ d \lfloor N^{1/d} \rfloor, N+1 \}$ and depth
 110 $12L + C_2$ such that*

$$111 \quad \|f - \phi\|_{L^p([0,1]^d)} \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}),$$

112 where $C_1 = 12$ and $C_2 = 14$ if $p \in [1, \infty)$; $C_1 = 3^{d+3}$ and $C_2 = 14 + 2d$ if $p = \infty$.

113 When Theorem 1.1 is applied to $f \in B_\lambda(C^\alpha([0, 1]^d))$, the approximation rate is
 114 $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$, because $\omega_f(r) \leq \lambda r^\alpha$ for any $r \geq 0$. An immediate question following
 115 the constructive approximation is how much we can improve the approximation rate. In
 116 fact, the approximation rate of $f \in B_\lambda(C^\alpha([0, 1]^d))$ is asymptotically tight based on
 117 VC-dimension as we shall see later.

118 In most real applications of neural networks, though the target function f is defined
 119 in a high-dimensional domain, e.g., $[0, 1]^d$, where d could be tens of thousands or even
 120 millions, only the approximation error of f in a neighborhood of a $d_{\mathcal{M}}$ -dimensional
 121 manifold \mathcal{M} with $d_{\mathcal{M}} \ll d$ is concerned. Hence, we extend Theorem 1.1 to the case
 122 when the domain of f is localized in an ε -neighborhood of a compact $d_{\mathcal{M}}$ -dimensional
 123 Riemannian submanifold $\mathcal{M} \subseteq [0, 1]^d$ having condition number $1/\tau$, volume V , and
 124 geodesic covering regularity \mathcal{R} . The ε -neighborhood is defined as

$$125 \quad \mathcal{M}_\varepsilon := \{ \mathbf{x} \in [0, 1]^d : \inf \{ \|\mathbf{x} - \mathbf{y}\|_2 : \mathbf{y} \in \mathcal{M} \} \leq \varepsilon \}, \quad \text{for } \varepsilon \in (0, 1). \quad (1.2)$$

126 Let $d_\delta = \mathcal{O}\left(\frac{d_{\mathcal{M}} \ln(dV\mathcal{R}\tau^{-1}\delta^{-1})}{\delta^2}\right) = \mathcal{O}\left(d_{\mathcal{M}} \frac{\ln(d/\delta)}{\delta^2}\right)$ be an integer for any $\delta \in (0, 1)$ such that
 127 $d_{\mathcal{M}} \leq d_\delta \leq d$. We show an approximation rate

$$128 \quad 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right)$$

129 for ReLU FNNs to pointwisely approximate f on \mathcal{M}_ε . The key ideas of the proof is the
 130 application of Theorem 3.1 in [3], which provides a nearly isometric projection $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$
 131 that maps points in $\mathcal{M} \subseteq [0, 1]^d$ to a d_δ -dimensional domain with

$$132 \quad (1 - \delta)\|\mathbf{x}_1 - \mathbf{x}_2\| \leq \|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2\| \leq (1 + \delta)\|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M},$$

133 and the application of Theorem 1.1 in this paper, which constructs the desired ReLU
 134 FNN with a size depending on d_δ instead of d to lessen the curse of dimensionality.
 135 When δ is closer to 1, d_δ is closer to $d_{\mathcal{M}}$ but the isometric property of the projection is
 136 weakened; when δ is closer to 0, the isometric property becomes better but d_δ could be
 137 larger than d , in which case we can simply enforce $d_\delta = d$ and choose the identity map
 138 as the projection. Hence, $\delta \in (0, 1)$ is a parameter to make a balance between isometry
 139 and dimension reduction.

140 **Theorem 1.2.** *Let f be a continuous function on $[0, 1]^d$ and $\mathcal{M} \subseteq [0, 1]^d$ be a com-*
 141 *pact $d_{\mathcal{M}}$ -dimensional Riemannian submanifold. For any $N \in \mathbb{N}^+$, $L \in \mathbb{N}^+$, $\varepsilon \in (0, 1)$,*
 142 *and $\delta \in (0, 1)$, there exists a function ϕ implemented by a ReLU FNN with width*
 143 *$3^{d_\delta+3} \max\{d_\delta \lfloor N^{1/d_\delta} \rfloor, N + 1\}$ and depth $12L + 14 + 2d_\delta$ such that*

$$144 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right), \quad (1.3)$$

145 for any $\mathbf{x} \in \mathcal{M}_\varepsilon$, where \mathcal{M}_ε is defined in Equation (1.2)

146 The approximation rate of deep neural networks for functions defined precisely on
 147 low-dimensional smooth manifolds has been studied in [60] for C^2 functions and in [9, 11]
 148 for Lipschitz continuous functions. Considering that it might be more reasonable to
 149 assume data located in a small neighborhood of low-dimensional smooth manifold in
 150 real applications, we introduce the ε -neighborhood of the manifold \mathcal{M} in Theorem 1.2.
 151 In general, existing results are again asymptotic and they cannot be applied to estimate
 152 the approximation accuracy of a ReLU FNN with arbitrarily given width N and depth L ,
 153 since there is no explicit formula without unknown constants to specify the exact error
 154 bound. For example, [9] provides an approximation rate $c_1(NL)^{-c_2/d_\delta}$ with unknown
 155 constants (e.g., c_1 and c_2) and requires NL greater than an unknown large number. The
 156 demand of an explicit error estimation motivates Theorem 1.2 in this paper. When data
 157 are concentrating around \mathcal{M} , ε is very small and the dominant term of the approximation
 158 error in (1.3) is $19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right)$ implying that the approximation via deep
 159 ReLU FNNs can lessen the curse of dimensionality.

160 The analysis above provides a general guide for selecting the width and depth of
 161 ReLU FNNs to approximate continuous functions, especially when the computation is
 162 conducted with parallel computing, which is usually the case in real applications [12, 58].
 163 As we shall see later, when the approximation accuracy and the parallel computing
 164 efficiency are considered together, very deep FNNs become less attractive than those
 165 with $\mathcal{O}(1)$ depth.

166 The approximation theories in this paper assume that the target function f is fully
 167 accessible, making it possible to estimate the approximation error and identify an asymp-
 168 totically optimal ReLU FNN with a given budget of neurons to minimize the approx-
 169 imation error. In real applications, usually only a limited number of possibly noisy
 170 observations of f is available, resulting in a regression problem in statistics. In the latter
 171 case, the problem is usually formulated in a stochastic setting with randomly generated
 172 noisy observations and the regression error contains mainly two components: bias and
 173 variance. The bias is the difference of the expectation of an estimated function and its

174 ground truth f . The approximation theories in this paper play an important role in
 175 characterizing the power of neural networks when they are applied to solve regression
 176 problems by providing a lower bound of the regression bias.

177 The rest of this paper is organized as follows. We first prove Theorem 1.1 and show
 178 its optimality in Section 2 when assuming Theorem 2.1 is true. Next, Theorem 2.1 is
 179 proved in Section 3. In Section 4, three aspects of neural networks in practice will be
 180 discussed: 1) neural network approximation in a high-dimensional irregular domain; 2)
 181 neural network approximation in the case of a low-dimensional data structure; 3) the
 182 optimal ReLU FNN in parallel computation. Finally, Section 5 concludes this paper
 183 with a short discussion.

184 2 Approximation of continuous functions

185 In this section, we prove Theorem 1.1 and discuss its optimality when assume The-
 186 orem 2.1 is true. Notations throughout the proof will be summarized in Section 2.1.

187 2.1 Notations

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

189 • Matrices are denoted by bold uppercase letters. For instance, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real
 190 matrix of size $m \times n$, and \mathbf{A}^T denotes the transpose of \mathbf{A} . Vectors are denoted
 191 as bold lowercase letters. For example, $\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} = [v_1, \dots, v_d]^T \in \mathbb{R}^d$ is a column
 192 vector with $\mathbf{v}(i) = v_i$ being the i -th element. Besides, “[” and “]” are used to
 193 partition matrices (vectors) into blocks, e.g., $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$.

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

$$195 \quad \|\mathbf{x}\|_p := (|x_1|^p + |x_2|^p + \dots + |x_d|^p)^{1/p}.$$

196 • Let $\mu(\cdot)$ be the Lebesgue measure.

197 • Let 1_S be the characteristic function on a set S , i.e., 1_S is equal to 1 on S and 0
 198 outside of S .

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

200 • For any $\xi \in \mathbb{R}$, let $\lfloor \xi \rfloor := \max\{i : i \leq \xi, i \in \mathbb{Z}\}$ and $\lceil \xi \rceil := \min\{i : i \geq \xi, i \in \mathbb{Z}\}$.

201 • Assume $\mathbf{n} \in \mathbb{N}^d$, then $f(\mathbf{n}) = \mathcal{O}(g(\mathbf{n}))$ means that there exists positive C indepen-
 202 dent of \mathbf{n} , f , and g such that $f(\mathbf{n}) \leq Cg(\mathbf{n})$ when all entries of \mathbf{n} go to $+\infty$.

203 • Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ denote the rectified linear unit (ReLU), i.e. $\sigma(x) = \max\{0, x\}$. With
 204 the abuse of notations, we define $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$ as $\sigma(\mathbf{x}) = \begin{bmatrix} \max\{0, x_1\} \\ \vdots \\ \max\{0, x_d\} \end{bmatrix}$ for any
 205 $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$.

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

207
$$\Omega([0, 1]^d, K, \delta) := \bigcup_{i=1}^d \left\{ \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_i \in \bigcup_{k=1}^{K-1} \left(\frac{k}{K} - \delta, \frac{k}{K} \right) \right\}. \quad (2.1)$$

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

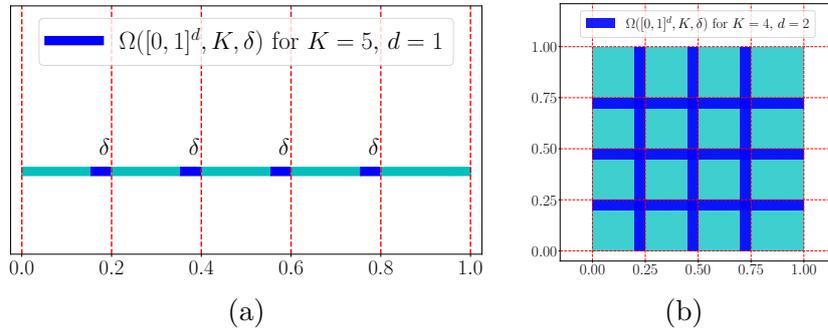


Figure 2: Two examples of trifling regions. (a) $K = 5, d = 1$. (b) $K = 4, d = 2$.

210 • Let $C^\alpha([0, 1]^d)$ be the set containing all Hölder continuous functions on $[0, 1]^d$ of order
 211 $\alpha \in (0, 1]$. In particular, the λ -ball in $C^\alpha([0, 1]^d)$ is denoted by $B_\lambda(C^\alpha([0, 1]^d))$
 212 for any $\lambda > 0$.

213 • We will use \mathcal{NN} to denote a function implemented by a ReLU FNN for short and use
 214 Python-type notations to specify a class of functions implemented by ReLU FNNs
 215 with several conditions, e.g., $\mathcal{NN}(c_1; c_2; \dots; c_m)$ is a set of functions implemented
 216 by ReLU FNNs satisfying m conditions given by $\{c_i\}_{1 \leq i \leq m}$, each of which may
 217 specify the number of inputs (#input), the number of outputs (#output), the
 218 total number of neurons in all hidden layers (#neuron), the number of hidden
 219 layers (depth), the total number of parameters (#parameter), and the width in
 220 each hidden layer (widthvec), the maximum width of all hidden layers (width),
 221 etc. For example, if $\phi \in \mathcal{NN}(\#input = 2; \text{widthvec} = [100, 100]; \#output = 1)$,
 222 then ϕ is a functions satisfies

- 223 – ϕ maps from \mathbb{R}^2 to \mathbb{R} .
- 224 – ϕ can be implemented by a ReLU FNN with two hidden layers and the number
 225 of nodes in each hidden layer is 100.

226 • $[n]^L$ is short for $[n, n, \dots, n] \in \mathbb{N}^L$. For example,

227
$$\mathcal{NN}(\#input = d; \text{widthvec} = [100, 100]) = \mathcal{NN}(\#input = d; \text{widthvec} = [100]^2).$$

- 228 • For a function $\phi \in \mathcal{NN}(\#\text{input} = d; \text{widthvec} = [N_1, N_2, \dots, N_L]; \#\text{output} = 1)$, if
 229 we set $N_0 = d$ and $N_{L+1} = 1$, then the architecture of the network implementing ϕ
 230 can be briefly described as follows:

231
$$\mathbf{x} = \tilde{\mathbf{h}}_0 \xrightarrow{\mathbf{W}_0, \mathbf{b}_0} \mathbf{h}_1 \xrightarrow{\sigma} \tilde{\mathbf{h}}_1 \cdots \xrightarrow{\mathbf{W}_{L-1}, \mathbf{b}_{L-1}} \mathbf{h}_L \xrightarrow{\sigma} \tilde{\mathbf{h}}_L \xrightarrow{\mathbf{W}_L, \mathbf{b}_L} \mathbf{h}_{L+1} = \phi(\mathbf{x}),$$

232 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
 233 the i -th (affine) linear transform \mathcal{L}_i in ϕ , respectively, i.e.,

234
$$\mathbf{h}_{i+1} = \mathbf{W}_i \cdot \tilde{\mathbf{h}}_i + \mathbf{b}_i =: \mathcal{L}_i(\tilde{\mathbf{h}}_i), \quad \text{for } i = 0, 1, \dots, L,$$

235 and

236
$$\tilde{\mathbf{h}}_i = \sigma(\mathbf{h}_i), \quad \text{for } i = 1, \dots, L.$$

237 In particular, ϕ can be represented in a form of function compositions as follows

238
$$\phi = \mathcal{L}_L \circ \sigma \circ \mathcal{L}_{L-1} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0,$$

which has been illustrated in Figure 3.

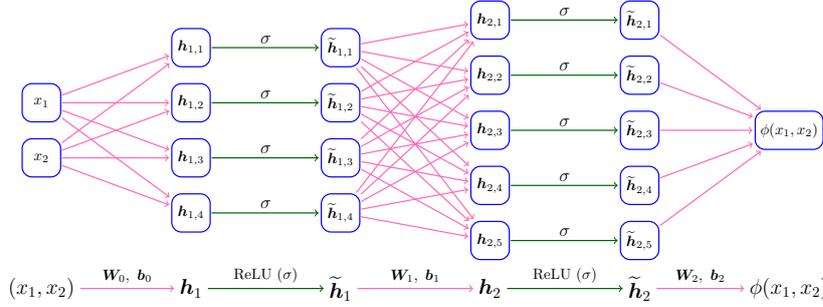


Figure 3: An example of a ReLU network with width 5 and depth 2.

239

- 240 • The expression “an FNN with width N and depth L ” means
- 241 – The maximum width of this FNN for all **hidden** layers is no more than N .
- 242 – The number of **hidden** layers of this FNN is no more than L .
- 243 • For $\theta \in [0, 1)$, suppose its binary representation is $\theta = \sum_{\ell=1}^{\infty} \theta_{\ell} 2^{-\ell}$ with $\theta_{\ell} \in \{0, 1\}$, we
 244 introduce a special notation $\text{bin}0.\theta_1\theta_2\cdots\theta_L$ to denote the L -term binary represen-
 245 tation of θ , i.e., $\text{bin}0.\theta_1\theta_2\cdots\theta_L := \sum_{\ell=1}^L \theta_{\ell} 2^{-\ell}$.

246 2.2 Proof of Theorem 1.1

247 We essentially construct piecewise constant functions to approximate continuous
 248 functions in the proof. However, it is impossible to construct a piecewise constant func-
 249 tion via ReLU FNNs due to the continuity of ReLU FNNs. Thus, we introduce the
 250 trifling region $\Omega([0, 1]^d, K, \delta)$, defined in Equation (2.1), and use ReLU FNNs to im-
 251 plement piecewise constant functions outside of the trifling region. To prove Theorem
 252 1.1, we first establish a theorem showing how to construct ReLU FNNs to pointwisely
 253 approximate continuous functions except for the trifling region.

254 **Theorem 2.1.** *Given $f \in C([0, 1]^d)$, for any $L \in \mathbb{N}^+$ and $N \in \mathbb{N}^+$, there exists a function*
 255 *ϕ implemented by a ReLU FNN with width $\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\}$ and depth*
 256 *$12L + 14$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and*

$$257 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

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

259 With Theorem 2.1 that will be proved in Section 3, we can easily prove Theorem
 260 1.1 for the case $p \in [1, \infty)$. In the early version of this paper, which focuses on contin-
 261 uous functions as target functions, we only considered the case $p \in [1, \infty)$ since it was
 262 challenging to control the approximation error in the trifling region. Later in [42] when
 263 we considered smooth functions as target functions, we invented a technique that can
 264 handle the error in the trifling region as in the lemma below. Therefore, we are now able
 265 to control the approximation error for $p = \infty$. The results in this paper are for continuous
 266 functions, to which the results in [42] are not applicable; the results in [42] characterize
 267 how the smoothness of target functions helps to enhance the approximation capacity of
 268 ReLU FNNs, which is not addressed in this paper. It is interesting to point out that the
 269 approximation rate $\mathcal{O}(N^{-2/d}L^{-2/d})$ for continuous functions in this paper is even better
 270 than the rate $\mathcal{O}((\frac{N}{\ln N})^{-2/d}(\frac{L}{\ln L})^{-2/d})$ for functions in $C^1([0, 1]^d)$ in [42].

271 **Lemma 2.2** (Theorem 2.1 of [42]). *Given $\varepsilon > 0$, $N, L, K \in \mathbb{N}^+$, and $\delta \in (0, \frac{1}{3K}]$, assume*
 272 *$f \in C([0, 1]^d)$ and $\tilde{\phi}$ can be implemented by a ReLU FNN with width N and depth L . If*

$$273 \quad |f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq \varepsilon, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

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

$$276 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

277 Now we are ready to prove Theorem 1.1 by assuming Theorem 2.1 is true, which
 278 will be proved later in Section 3.2.

279 *Proof of Theorem 1.1.* Let us first consider the case $p \in [1, \infty)$. We may assume f is
 280 not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. Set
 281 $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

$$282 \quad Kd\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor d\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p \\ \leq (\omega_f(N^{-2/d}L^{-2/d}))^p.$$

283 By Theorem 2.1, there exists a function ϕ implemented by a ReLU FNN with width

$$284 \quad \max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\} \leq 12 \max\{d\lfloor N^{1/d} \rfloor, N + 1\}$$

285 and depth $12L + 14$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and

$$286 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

287 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

$$\begin{aligned}
\|f - \phi\|_{L^p([0,1]^d)}^p &= \int_{\Omega([0,1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} + \int_{[0,1]^d \setminus \Omega([0,1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} \\
&\leq Kd\delta(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d}))^p + (18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p \\
&\leq (\omega_f(N^{-2/d}L^{-2/d}))^p + (18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p \\
&\leq (19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p.
\end{aligned}$$

288

289 Hence, $\|f - \phi\|_{L^p([0,1]^d)} \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$.

290 Next, let us discuss the case $p = \infty$. Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and choose a small
291 $\delta \in (0, \frac{1}{3K}]$ such that

$$292 \quad d \cdot \omega_f(\delta) \leq \omega_f(N^{-2/d}L^{-2/d}).$$

293 By Theorem 2.1, there exists a function $\tilde{\phi}$ implemented by a ReLU FNN with width
294 $\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\}$ and depth $12L + 14$ such that

$$295 \quad |f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}) := \varepsilon, \quad \text{for } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

296 By Lemma 2.2, there exists a function ϕ implemented by a ReLU FNN with width

$$297 \quad 3^d \left(\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\} + 4 \right) \leq 3^{d+3} \max\{d\lfloor N^{1/d} \rfloor, N + 1\}$$

298 and depth $12L + 14 + 2d$ such that

$$299 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta) \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

300 So we finish the proof. □

301 2.3 Optimality of Theorem 1.1

302 This section will show that the approximation rate in Theorem 1.1 is nearly tight
303 and there is no room to improve for the function class $B_\lambda(C^\alpha([0, 1]^d))$. Theorem 2.3
304 below shows that the approximation rate $\mathcal{O}(\omega_f(N^{-(2/d+\rho)}L^{-(2/d+\rho)}))$ for any $\rho > 0$ is
305 unachievable, implying the approximation rate in Theorem 1.1 is nearly tight for the
306 function class $B_\lambda(C^\alpha([0, 1]^d))$.

307 **Theorem 2.3.** *Given any $\rho > 0$ and $C > 0$, there exists $f \in B_\lambda(C^\alpha([0, 1]^d))$ such that,*
308 *for any $J_0 > 0$, there exist $N, L \in \mathbb{N}$ with $NL \geq J_0$ satisfying*

$$309 \quad \inf_{\phi \in \mathcal{NN}(\#\text{input}=d; \text{width} \leq N; \text{depth} \leq L)} \|\phi - f\|_{L^\infty([0,1]^d)} \geq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}.$$

310 In fact, we can show a stronger result than Theorem 2.3. Under the same con-
311 ditions as in Theorem 2.3, for any $\mathcal{H} \in [0, 1]^d$ with $\mu(\mathcal{H}) \leq 2^{-(d+K^{d+1})}K^{-d}$, where $K =$
312 $\lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor$, it can be proved that

$$313 \quad \inf_{\phi \in \mathcal{NN}(\#\text{input}=d; \text{width} \leq N; \text{depth} \leq L)} \|\phi - f\|_{L^\infty([0,1]^d \setminus \mathcal{H})} \geq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}. \quad (2.2)$$

314 We will prove (2.2) by contradiction, then Theorem 2.3 holds as a consequence. Assuming
315 Equation (2.2) is false, we have the following claim.

316 **Claim 2.4.** *There exist $\rho > 0$ and $C > 0$ such that given any $f \in B_\lambda(C^\alpha([0, 1]^d))$,*
 317 *there exists $J_0 = J_0(\rho, C, f) > 0$ such that, for any $N, L \in \mathbb{N}$ with $NL \geq J_0$, there exist*
 318 *$\phi \in \mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)$ and $\mathcal{H} \in [0, 1]^d$ with $\mu(\mathcal{H}) \leq 2^{-(d+K^d+1)}K^{-d}$,*
 319 *where $K = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor$, satisfying*

$$320 \quad \|f - \phi\|_{L^\infty([0,1]^d \setminus \mathcal{H})} \leq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}.$$

321 Now let us disprove this claim to show Theorem 2.3 and Equation (2.2) are true.

322 *Disproof of Claim 2.4.* Without the loss of generality, we assume $\lambda = 1$; in the case of
 323 $\lambda \neq 1$, the proof is similar. We will disprove Claim 2.4 using the VC dimension. Recall
 324 that the VC dimension of a class of functions is defined as the cardinality of the largest
 325 set of points that this class of functions can shatter. Denote the VC dimension of a
 326 function set \mathcal{F} by $\text{VCDim}(\mathcal{F})$. By [27] and the fact

$$327 \quad \mathcal{NN}(\text{width} \leq N; \text{depth} \leq L) \subseteq \mathcal{NN}(\#\text{parameter} \leq (LN + d + 2)(N + 1)),$$

328 there exists $C_1 > 0$ such that

$$329 \quad \begin{aligned} & \text{VCDim}(\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)) \\ & \leq C_1(LN + d + 2)(N + 1)L \ln((LN + d + 2)(N + 1)) \\ & =: b_u(N, L). \end{aligned} \quad (2.3)$$

330 Then we will use Claim 2.4 to estimate a lower bound of

$$331 \quad \text{VCDim}(\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)), \quad (2.4)$$

332 and this lower bound is asymptotically larger than $b_u(N, L)$, which leads to a contradic-
 333 tion.

334 More precisely, we will construct $\{f_\chi : \chi \in \mathcal{B}\} \subseteq B_1(C^\alpha([0, 1]^d))$, which can shat-
 335 ter $b_\ell(N, L) := K^d$ points, where \mathcal{B} is a set defined later. Then by Claim 2.4, there
 336 exists $\{\phi_\chi : \chi \in \mathcal{B}\}$ such that this set can shatter $b_\ell(N, L)$ points. Finally, $b_\ell(N, L) =$
 337 $K^d = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^d$ is asymptotically larger than $b_u(N, L) = C_1(LN + d + 2)(N +$
 338 $1)L \ln((LN + d + 2)(N + 1))$, which leads to a contradiction. More details can be found
 339 below.

340 **Step 1:** Construct $\{f_\chi : \chi \in \mathcal{B}\} \subseteq B_1(C^\alpha([0, 1]^d))$ that scatters $b_\ell(N, L)$ points.

341 Divide $[0, 1]^d$ into K^d non-overlapping sub-cubes $\{Q_\beta\}_\beta$ as follows:

$$342 \quad Q_\beta := \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_i \in [\frac{\beta_i-1}{K}, \frac{\beta_i}{K}], i = 1, 2, \dots, d\},$$

343 for any index vector $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T \in \{1, 2, \dots, K\}^d$.

344 Let $Q(\mathbf{x}_0, \eta) \subseteq [0, 1]^d$ be a hypercube, whose center and sidelength are \mathbf{x}_0 and η ,
 345 respectively. Then we define a function ζ_Q on $[0, 1]^d$ corresponding to $Q = Q(\mathbf{x}_0, \eta) \subseteq$
 346 $[0, 1]^d$ such that:

$$347 \quad \bullet \zeta_Q(\mathbf{x}_0) = (\eta/2)^\alpha/2;$$

- 348 • $\zeta_Q(\mathbf{x}) = 0$ for any $\mathbf{x} \notin Q \setminus \partial Q$, where ∂Q is the boundary of Q ;
 349 • ζ_Q is linear on the line that connects \mathbf{x}_0 and \mathbf{x} , for any $\mathbf{x} \in \partial Q$.

350 Define

351
$$\mathcal{B} := \{\chi : \chi \text{ is a map from } \{1, 2, \dots, K\}^d \text{ to } \{-1, 1\}\}.$$

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

353
$$f_\chi(\mathbf{x}) := \sum_{\beta \in \{1, 2, \dots, K\}^d} \chi(\beta) \zeta_{Q_\beta}(\mathbf{x}),$$

354 where $\zeta_{Q_\beta}(\mathbf{x})$ is the associated function introduced just above. It is easy to check that
 355 $\{f_\chi : \chi \in \mathcal{B}\} \subseteq B_1(C^\alpha([0, 1]^d))$ can shatter $b_\ell(N, L) = K^d$ points.

356 **Step 2:** Construct $\{\phi_\chi : \chi \in \mathcal{B}\}$ that scatters $b_\ell(N, L)$ points.

357 By Claim 2.4, there exist $\rho > 0$ and $C_2 > 0$ such that, for any $f_\chi \in \{f_\chi : \chi \in \mathcal{B}\}$ there
 358 exists $J_\chi > 0$ such that for all $N, L \in \mathbb{N}$ with $NL \geq J_\chi$, there exist $\phi_\chi \in \mathcal{NN}(\#input =$
 359 $d; \text{ width} \leq N; \text{ depth} \leq L)$ and \mathcal{H}_χ with $\mu(\mathcal{H}_\chi) \leq 2^{-(d+K^d+1)}K^{-d}$ such that

360
$$|f_\chi(\mathbf{x}) - \phi_\chi(\mathbf{x})| \leq C_2(NL)^{-\alpha(2/d+\rho/\alpha)}, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}_\chi.$$

361 Set $\mathcal{H} = \cup_{\chi \in \mathcal{B}} \mathcal{H}_\chi$ and $J_1 = \max_{\chi \in \mathcal{B}} J_\chi$. Then it holds that

362
$$\mu(\mathcal{H}) \leq 2^{K^d} 2^{-(d+K^d+1)} K^{-d} = (2K)^{-d}/2. \quad (2.5)$$

363 It follows that for all $\chi \in \mathcal{B}$ and $N, L \in \mathbb{N}$ with $NL \geq J_1$, we have

364
$$|f_\chi(\mathbf{x}) - \phi_\chi(\mathbf{x})| \leq C_2(NL)^{-\alpha(2/d+\rho/\alpha)}, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}. \quad (2.6)$$

365 For each index vector $\beta \in \{1, 2, \dots, K\}^d$ and any $\mathbf{x} \in \frac{1}{2}Q_\beta$, where $\frac{1}{2}Q_\beta$ denotes the
 366 cube whose sidelength is half of that of Q_β sharing the same center of Q_β , since Q_β has
 367 a sidelength $\frac{1}{K} = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^{-1}$, we have

368
$$|f_\chi(\mathbf{x})| = |\zeta_{Q_\beta}(\mathbf{x})| \geq |\zeta_{Q_\beta}(\mathbf{x}_{Q_\beta})|/2 = \left(\frac{1}{2K}\right)^\alpha / 4 = \frac{1}{2^{2+\alpha}} \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^{-\alpha}, \quad (2.7)$$

369 where \mathbf{x}_{Q_β} is the center of Q_β . For fixed d, α , and ρ , there exists $J_2 > 0$ large enough
 370 such that, for any $N, L \in \mathbb{N}$ with $NL \geq J_2$, we have

371
$$\frac{1}{2^{2+\alpha}} \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^{-\alpha} > C_2(NL)^{-\alpha(2/d+\rho/\alpha)}. \quad (2.8)$$

372 By Equation (2.5), for any $\beta \in \{1, 2, \dots, K\}^d$, we have

373
$$\mu(\mathcal{H}) \leq (2K)^{-d}/2 < (2K)^{-d} = \mu(\frac{1}{2}Q_\beta),$$

374 which means $(\frac{1}{2}Q_\beta) \cap ([0, 1]^d \setminus \mathcal{H})$ is not empty. Therefore, there exists $\mathbf{x}_\beta \in (\frac{1}{2}Q_\beta) \cap$
 375 $([0, 1]^d \setminus \mathcal{H})$ for each $\beta \in \{1, 2, \dots, K\}^d$ such that

376
$$|f_\chi(\mathbf{x}_\beta)| \geq \frac{1}{2^{2+\alpha}} \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^{-\alpha} > C_2(NL)^{-\alpha(2/d+\rho/\alpha)} \geq |f_\chi(\mathbf{x}_\beta) - \phi_\chi(\mathbf{x}_\beta)|,$$

377 for any $N, L \in \mathbb{N}$ with $NL \geq J_0 = \max\{J_1, J_2\}$, where the first, the second, and the
 378 last inequalities come from (2.7), (2.8), and (2.6), respectively. In other words, for any
 379 $\chi \in \mathcal{B}$ and $\beta \in \{1, 2, \dots, K\}^d$, $f_\chi(\mathbf{x}_\beta)$ and $\phi_\chi(\mathbf{x}_\beta)$ have the same sign. Then $\{\phi_\chi : \chi \in \mathcal{B}\}$
 380 shatters $\{\mathbf{x}_\beta : \beta \in \{1, 2, \dots, K\}^d\}$ since $\{f_\chi : \chi \in \mathcal{B}\}$ shatters $\{\mathbf{x}_\beta : \beta \in \{1, 2, \dots, K\}^d\}$ as
 381 discussed in Step 1. Hence,

$$382 \quad \text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \geq K^d = b_\ell(N, L), \quad (2.9)$$

383 for any $N, L \in \mathbb{N}$ with $NL \geq J_0$,

384 **Step 3:** Contradiction.

385 By Equation (2.3) and (2.9), for any $N, L \in \mathbb{N}$ with $NL \geq J_0$, we have

$$386 \quad \begin{aligned} b_\ell(N, L) &\leq \text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \\ &\leq \text{VCDim}(\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)) \leq b_u(N, L), \end{aligned}$$

387 implying that

$$388 \quad [(NL)^{2/d+\rho/(2\alpha)}]^d \leq C_1(LN + d + 2)(N + 1)L \ln((LN + d + 2)(N + 1)),$$

389 which is a contradiction for sufficiently large $N, L \in \mathbb{N}$. So we finish the proof. \square

390 By Theorem 2.3, for any $\rho > 0$, the approximation rate cannot be better than
 391 $\mathcal{O}(N^{-(2\alpha/d+\rho)}L^{-(2/\alpha+\rho)})$, if we use FNNs in $\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)$ to
 392 approximate functions in $B_\lambda(C^\alpha([0, 1]^d))$. By a similar argument, we can show that the
 393 approximation rate cannot be $\mathcal{O}(N^{-2\alpha/d}L^{-(2/\alpha+\rho)})$ nor $\mathcal{O}(N^{-(2\alpha/d+\rho)}L^{-2\alpha/d})$. Hence, the
 394 approximation rate in Theorem 1.1 is nearly tight.

395 **3 Proof of Theorem 2.1**

396 In this section, we will prove Theorem 2.1. We first present the key ideas in Section
 397 3.1. Based on two propositions in Section 3.1, the detailed proof is presented in Section
 398 3.2. Finally, the proofs of two propositions in Section 3.1 can be found in Section 3.3
 399 and 3.4.

400 **3.1 Key ideas of proving Theorem 2.1**

401 We will show that an almost piecewise constant function ϕ implemented by a ReLU
 402 FNN is enough to achieve the desired approximation rate in Theorem 1.1. Given an
 403 arbitrary $f \in C([0, 1]^d)$, we introduce a piecewise constant function $f_p \approx f$ serving as an
 404 intermediate approximant in our construction in the sense that

$$405 \quad f \approx f_p \text{ on } [0, 1]^d, \quad \text{and} \quad f_p \approx \phi \text{ on } [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta).$$

406 The approximation in $f \approx f_p$ is a simple and standard technique in constructive approxi-
 407 mation. For example, given arbitrary N and L , uniformly partition $[0, 1]^d$ into $\mathcal{O}(N^2L^2)$

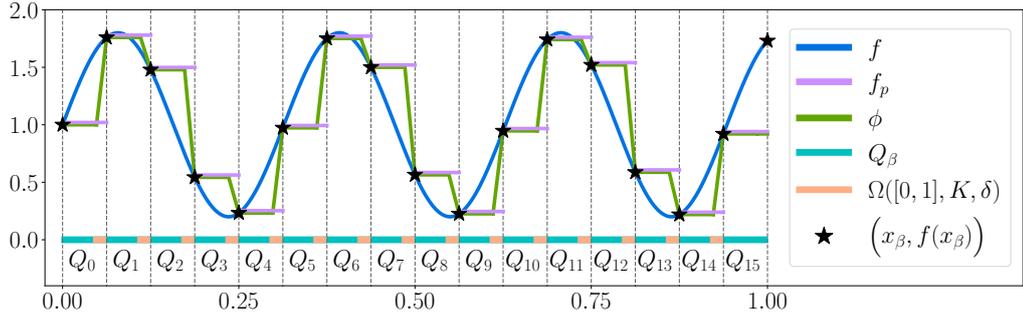


Figure 4: An illustration of f , f_p , ϕ , x_β , Q_β , and the trifling region $\Omega([0, 1]^d, K, \delta)$ in the one-dimensional case for $\beta \in \{0, 1, \dots, K-1\}^d$, where $K = N^2 L^2$ and $d = 1$ with $N = 2$ and $L = 2$. f is the target function; f_p is the piecewise constant function approximating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

408 pieces and define f_p using this partition. Then the approximation error of $f_p \approx f$ scales
409 like $\mathcal{O}(N^{-2/d} L^{-2/d})$. We will address the approximation in $f_p \approx \phi$ with the same error
410 scaling and a limited budget of the FNN size, e.g., $\mathcal{O}(NL)$ neurons, based on the fact
411 that f_p can be approximately implemented by a ReLU FNN in $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$,
412 where $\Omega([0, 1]^d, K, \delta)$ is the trifling region near the discontinuous locations of f_p with an
413 arbitrarily small Lebesgue measure (see Figure 4 for an illustration). The introduction
414 of the trifling region is to ease the construction of a deep ReLU FNN to implement the
415 desired ϕ , which is a piecewise linear and continuous function, to approximate the dis-
416 continuous function f_p by removing the difficulty near discontinuous points, essentially
417 smoothing f_p by restricting the approximation domain in $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$.

418 Now let us discuss the detailed steps of construction. First, divide $[0, 1]^d$ into a union
419 of important regions $\{Q_\beta\}_\beta$ and the trifling region $\Omega([0, 1]^d, K, \delta)$, where each Q_β is
420 associated with a representative $x_\beta \in Q_\beta$ such that $f(x_\beta) = f_p(x_\beta)$ for each index vector
421 $\beta \in \{0, 1, \dots, K-1\}^d$, where $K = \mathcal{O}(N^{2/d} L^{2/d})$ is the partition number per dimension
422 (see Figure 6 for examples for $d = 1$ and $d = 2$). Next, we design a vector function
423 $\Phi_1(\mathbf{x})$ constructed via $\Phi_1(\mathbf{x}) = [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T$ to project the whole cube
424 Q_β to a d -dimensional index β for each β , where each one-dimensional function ϕ_1 is
425 a step function implemented by a ReLU FNN. The final step is to solve a point fitting
426 problem. To be precise, we construct a function ϕ_2 implemented by a ReLU FNN to
427 map β approximately to $f_p(x_\beta) = f(x_\beta)$. Then $\phi_2 \circ \Phi_1(\mathbf{x}) = \phi_2(\beta) \approx f_p(x_\beta) = f(x_\beta)$
428 for any $\mathbf{x} \in Q_\beta$ and each β , implying $\phi := \phi_2 \circ \Phi_1 \approx f_p \approx f$ on $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$.
429 We would like to point out that we only need to care about the values of ϕ_2 at a set
430 of points $\{0, 1, \dots, K-1\}^d$ in the construction of ϕ_2 according to our design $\phi = \phi_2 \circ \Phi_1$
431 as illustrated in Figure 5. Therefore, it is unnecessary to care about the values of ϕ_2
432 sampled outside the set $\{0, 1, \dots, K-1\}^d$, which is a key point to ease the design of a
433 ReLU FNN to implement ϕ_2 as we shall see later.

434 Finally, we discuss how to implement Φ_1 and ϕ_2 by deep ReLU FNNs with width

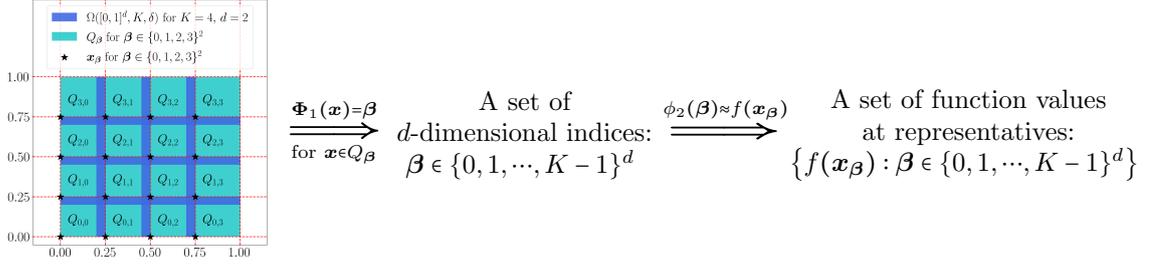


Figure 5: An illustration of the desired function $\phi = \phi_2 \circ \Phi_1$. Note that $\phi \approx f$ on $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$, since $\phi(\mathbf{x}) = \phi_2 \circ \Phi_1(\mathbf{x}) = \phi_2(\beta) \approx f(\mathbf{x}_\beta)$ for any $\mathbf{x} \in Q_\beta$ and each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

435 $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ using two propositions as we shall prove in Section 3.3 and 3.4
 436 later. We first construct a ReLU FNN with desired width and depth by Proposition 3.1
 437 to implement a one-dimensional step function ϕ_1 . Then Φ_1 can be attained via defining

$$438 \quad \Phi_1(\mathbf{x}) = [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d.$$

439 **Proposition 3.1.** For any $N, L, d \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{3K}]$ with $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$, there
 440 exists a one-dimensional function ϕ implemented by a ReLU FNN with width $4 \lfloor N^{1/d} \rfloor + 3$
 441 and depth $4L + 5$ such that

$$442 \quad \phi(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot \mathbf{1}_{\{k \leq K-2\}} \right] \text{ for } k = 0, 1, \dots, K-1.$$

443 The construction of ϕ_2 is a direct result of Proposition 3.2 below, the proof of which
 444 relies on the bit extraction technique in [5].

445 **Proposition 3.2.** Given any $\varepsilon > 0$ and arbitrary $N, L, J \in \mathbb{N}^+$ with $J \leq N^2 L^2$, assume
 446 $\{y_j \geq 0 : j = 0, 1, \dots, J-1\}$ is a sample set with $|y_j - y_{j-1}| \leq \varepsilon$ for $j = 1, 2, \dots, J-1$. Then
 447 there exists $\phi \in \mathcal{NN}(\#input = 1; \text{width} \leq 12N + 8; \text{depth} \leq 4L + 9; \#output = 1)$ such
 448 that

- 449 (i) $|\phi(j) - y_j| \leq \varepsilon$ for $j = 0, 1, \dots, J-1$;
 450 (ii) $0 \leq \phi(x) \leq \max\{y_j : j = 0, 1, \dots, J-1\}$ for any $x \in \mathbb{R}$.

451 With the above propositions ready, let us prove Theorem 2.1 in Section 3.2. We
 452 further assume that $\omega_f(r) > 0$ for any $r > 0$, excluding a simple case when f is a constant
 453 function.

454 3.2 Proof of Theorem 2.1

455 We essentially construct an almost piecewise constant function implemented by a
 456 ReLU FNN with $\mathcal{O}(NL)$ neurons to approximate f . We may f is not a constant since
 457 it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. It is clear that $|f(\mathbf{x}) - f(\mathbf{0})| \leq \omega_f(\sqrt{d})$
 458 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
 459 $\mathbf{x} \in [0, 1]^d$. Let $M = N^2 L$, $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$, and δ be an arbitrary number in $(0, \frac{1}{3K}]$.

460 The proof can be divided into four steps as follows:

- 461 1. Divide $[0, 1]^d$ into a union of sub-cubes $\{Q_\beta\}_{\beta \in \{0, 1, \dots, K-1\}^d}$ and the trifling region
462 $\Omega([0, 1]^d, K, \delta)$, and denote \mathbf{x}_β as the vertex of Q_β with minimum $\|\cdot\|_1$ norm;
- 463 2. Construct a sub-network to implement a vector function Φ_1 projecting the whole
464 cube Q_β to the d -dimensional index β for each β , i.e., $\Phi_1(\mathbf{x}) = \beta$ for all $\mathbf{x} \in Q_\beta$;
- 465 3. Construct a sub-network to implement a function ϕ_2 mapping the index β approx-
466 imately to $\tilde{f}(\mathbf{x}_\beta)$. This core step can be further divided into three sub-steps:
- 467 3.1. Construct a sub-network to implement ψ_1 bijectively mapping the index set
468 $\{0, 1, \dots, K-1\}^d$ to an auxiliary set $\mathcal{A}_1 \subseteq \{\frac{j}{2K^d} : j = 0, 1, \dots, 2K^d\}$ defined later
469 (see Figure 7 for an illustration);
- 470 3.2. Determine a continuous piecewise linear function g with a set of breakpoints
471 $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ satisfying: 1) assign the values of g at breakpoints in \mathcal{A}_1 based
472 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
473 in $\mathcal{A}_2 \cup \{1\}$ to reduce the variation of g for applying Proposition 3.2;
- 474 3.3. Apply Proposition 3.2 to construct a sub-network to implement a function ψ_2
475 approximating g well on $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$. Then the desired function ϕ_2 is given
476 by $\phi_2 = \psi_2 \circ \psi_1$ satisfying $\phi_2(\beta) = \psi_2 \circ \psi_1(\beta) \approx g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$;
- 477 4. Construct the final target network to implement the desired function ϕ such that
478 $\phi(\mathbf{x}) = \phi_2 \circ \Phi_1(\mathbf{x}) + f(\mathbf{0}) - \omega_f(\sqrt{d}) \approx \tilde{f}(\mathbf{x}_\beta) + f(\mathbf{0}) - \omega_f(\sqrt{d}) = f(\mathbf{x}_\beta)$ for $\mathbf{x} \in Q_\beta$.

479 The details of these steps can be found below.

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

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

$$482 \quad Q_\beta := \left\{ \mathbf{x} = [x_1, \dots, x_d]^T \in [0, 1]^d : x_i \in \left[\frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \leq K-2\}} \right], i = 1, \dots, d \right\}$$

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

$$486 \quad [0, 1]^d = \left(\cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta \right) \cup \Omega([0, 1]^d, K, \delta),$$

487 see Figure 6 for illustrations.

488 **Step 2:** Construct Φ_1 mapping $\mathbf{x} \in Q_\beta$ to β .

489 By Proposition 3.1, there exists $\phi_1 \in \mathcal{NN}$ (width $\leq 4\lfloor N^{1/d} \rfloor + 3$; depth $\leq 4L + 5$) such
490 that

$$491 \quad \phi_1(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}} \right] \text{ for } k = 0, 1, \dots, K-1.$$

492 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 $\beta = [\beta_1, \beta_2, \dots, \beta_d]^T$.

493 By defining

$$494 \quad \Phi_1(\mathbf{x}) := [\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)]^T, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d,$$

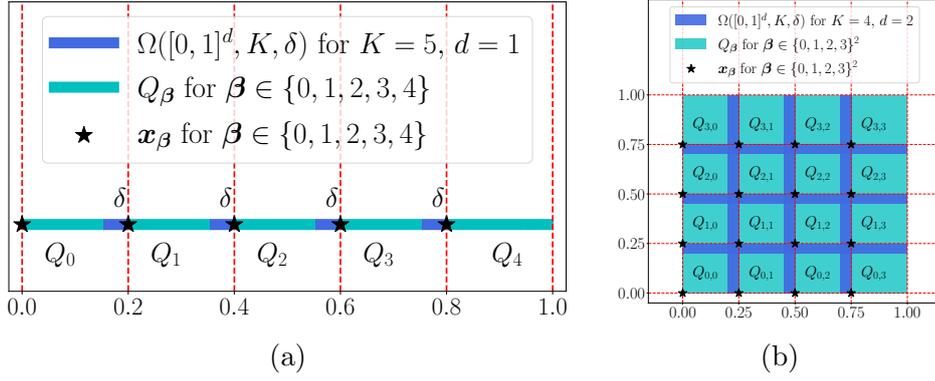


Figure 6: Illustrations of $\Omega([0, 1]^d, K, \delta)$, Q_β , and \mathbf{x}_β for $\beta \in \{0, 1, \dots, K-1\}^d$. (a) $K = 5$ and $d = 1$. (b) $K = 4$ and $d = 2$.

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

496 **Step 3:** Construct ϕ_2 mapping β approximately to $\tilde{f}(\mathbf{x}_\beta)$.

497 The construction of the sub-network implementing ϕ_2 is essentially based on Propo-
 498 sition 3.2. To meet the requirements of applying Proposition 3.2, we first define two
 499 auxiliary set \mathcal{A}_1 and \mathcal{A}_2 as

$$500 \quad \mathcal{A}_1 := \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\}$$

501 and

$$502 \quad \mathcal{A}_2 := \left\{ \frac{i}{K^{d-1}} + \frac{K+k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\}.$$

503 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 6 for an
 504 illustration of \mathcal{A}_1 and \mathcal{A}_2 . Next, we further divide this step into three sub-steps.

505 **Step 3.1:** Construct ψ_1 bijectively mapping $\{0, 1, \dots, K-1\}^d$ to \mathcal{A}_1 .

506 Inspired by the binary representation, we define

$$507 \quad \psi_1(\mathbf{x}) := \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d. \quad (3.1)$$

508 Then ψ_1 is a linear function bijectively mapping the index set $\{0, 1, \dots, K-1\}^d$ to

$$509 \quad \left\{ \frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i} : \beta \in \{0, 1, \dots, K-1\}^d \right\} \\ = \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1}-1 \quad \text{and} \quad k = 0, 1, \dots, K-1 \right\} = \mathcal{A}_1.$$

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

512 Let $g : [0, 1] \rightarrow \mathbb{R}$ be a continuous piecewise linear function with a set of breakpoints
 513 $\left\{ \frac{j}{2K^d} : j = 0, 1, \dots, 2K^d \right\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ and the values of g at these breakpoints satisfy
 514 the following properties:

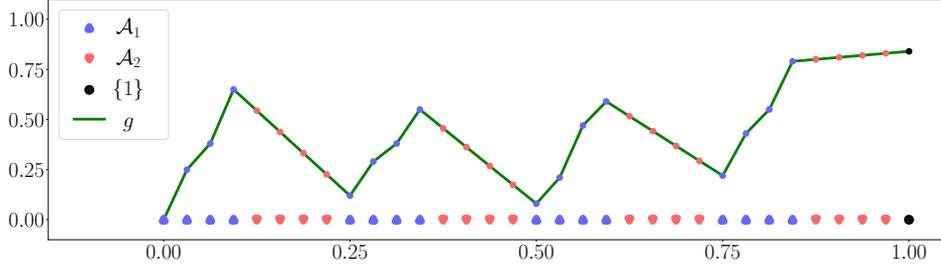


Figure 7: An illustration of \mathcal{A}_1 , \mathcal{A}_2 , $\{1\}$, and g for $d = 2$ and $K = 4$.

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

516
$$g(\psi_1(\boldsymbol{\beta})) = \tilde{f}(\mathbf{x}_\beta), \quad \text{for any } \boldsymbol{\beta} \in \{0, 1, \dots, K-1\}^d; \quad (3.2)$$

- 517 • At the breakpoint 1, let $g(1) = \tilde{f}(\mathbf{1})$, where $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathbb{R}^d$;

- 518 • The values of g at the breakpoints in \mathcal{A}_2 are assigned to reduce the variation of g ,
519 which is a requirement of applying Proposition 3.2. Note that

520
$$\left\{ \frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}} \right\} \subseteq \mathcal{A}_1 \cup \{1\}, \quad \text{for } i = 1, 2, \dots, K^{d-1},$$

521 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 $i = 1, 2, \dots, K^{d-1}$.

522 Thus, the values of g at the breakpoints in \mathcal{A}_2 can be successfully assigned by
523 letting g linear on each interval $[\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}}]$ for $i = 1, 2, \dots, K^{d-1}$, since
524 $\mathcal{A}_2 \subseteq \cup_{i=1}^{K^{d-1}} [\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}}]$.

525 Apparently, such a function g exists (see Figure 7 for an example) and satisfies

526
$$\left| g\left(\frac{j}{2K^d}\right) - g\left(\frac{j-1}{2K^d}\right) \right| \leq \max \left\{ \omega_f\left(\frac{1}{K}\right), \omega_f(\sqrt{d})/K \right\} \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \quad \text{for } j = 1, 2, \dots, 2K^d,$$

527 and

528
$$0 \leq g\left(\frac{j}{2K^d}\right) \leq 2\omega_f(\sqrt{d}), \quad \text{for } j = 0, 1, \dots, 2K^d.$$

529 **Step 3.3:** Construct ψ_2 approximating g well on $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$.

530 Since $2K^d = 2(\lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor)^d \leq 2(N^2 L^2) \leq N^2 \tilde{L}^2$, where $\tilde{L} = 2L$, by Proposition 3.2
531 (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; width \leq
532 $12N + 8$; depth $\leq 4\tilde{L} + 9$) = \mathcal{NN} (#input = 1; width $\leq 12N + 8$; depth $\leq 8L + 9$) such that

533
$$|\tilde{\psi}_2(j) - g\left(\frac{j}{2K^d}\right)| \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \quad \text{for } j = 0, 1, \dots, 2K^d - 1,$$

534 and

535
$$0 \leq \tilde{\psi}_2(x) \leq \max \left\{ g\left(\frac{j}{2K^d}\right) : j = 0, 1, \dots, 2K^d - 1 \right\} \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}.$$

536 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; width \leq
537 $12N + 8$; depth $\leq 8L + 9$),

538
$$0 \leq \psi_2(x) = \tilde{\psi}_2(2K^d x) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}, \quad (3.3)$$

539 and

$$540 \quad |\psi_2(\frac{j}{2K^d}) - g(\frac{j}{2K^d})| = |\tilde{\psi}_2(j) - g(\frac{j}{2K^d})| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad \text{for } j = 0, 1, \dots, 2K^d - 1. \quad (3.4)$$

541 Let us end Step 3 by defining the desired function ϕ_2 as $\phi_2 := \psi_2 \circ \psi_1$. Note that $\psi_1 : \mathbb{R}^d \rightarrow \mathbb{R}$ is a linear function and $\psi_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 12N + 8; \text{depth} \leq 8L + 9)$.
 542 Thus, $\phi_2 \in \mathcal{NN}(\#input = d; \text{width} \leq 12N + 8; \text{depth} \leq 8L + 9)$. By Equation (3.2) and
 543 (3.4), we have

$$545 \quad |\phi_2(\boldsymbol{\beta}) - \tilde{f}(\mathbf{x}_\beta)| = |\psi_2(\psi_1(\boldsymbol{\beta})) - g(\psi_1(\boldsymbol{\beta}))| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad (3.5)$$

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

$$547 \quad 0 \leq \phi_2(\mathbf{x}) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d. \quad (3.6)$$

548 **Step 4:** Construct the final network to implement the desired function ϕ .

549 Define $\phi := \phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$. Since $\phi_1 \in \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq$
 550 $4L + 5]$), we have $\Phi_1 \in \mathcal{NN}(\#input = d; \text{width} \leq 4d\lfloor N^{1/d} \rfloor + 3d; \text{depth} \leq 4L + 5; \#output =$
 551 $d)$. Note that $\phi_2 \in \mathcal{NN}(\#input = d; \text{width} \leq 12N + 8; \text{depth} \leq 8L + 9)$. Thus, $\phi =$
 552 $\phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$ is in

$$553 \quad \mathcal{NN}(\text{width} \leq \max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\}; \text{depth} \leq (4L + 5) + (8L + 9) = 12L + 14).$$

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

$$556 \quad \begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &= |\tilde{f}(\mathbf{x}) - \phi_2(\Phi_1(\mathbf{x}))| = |\tilde{f}(\mathbf{x}) - \phi_2(\boldsymbol{\beta})| \\ &\leq |\tilde{f}(\mathbf{x}) - \tilde{f}(\mathbf{x}_\beta)| + |\tilde{f}(\mathbf{x}_\beta) - \phi_2(\boldsymbol{\beta})| \\ &\leq \omega_f(\frac{\sqrt{d}}{K}) + \omega_f(\frac{\sqrt{d}}{K}) \leq 2\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d}), \end{aligned}$$

557 where the last inequality comes from the fact $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor \geq \frac{N^{2/d}L^{2/d}}{8}$ for any $N, L \in$
 558 \mathbb{N}^+ . Recall the fact $\omega_f(nr) \leq n\omega_f(r)$ for any $n \in \mathbb{N}^+$ and $r \in [0, \infty)$. Therefore, for any
 559 $\mathbf{x} \in \cup_{\boldsymbol{\beta} \in \{0, 1, \dots, K-1\}^d} Q_\beta = [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$, we have

$$560 \quad \begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq 2\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d}) \leq 2\lceil 8\sqrt{d} \rceil \omega_f(N^{-2/d}L^{-2/d}) \\ &\leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}). \end{aligned}$$

561 It remains to show the upper bound of ϕ . By Equation (3.6) and $\phi = \phi_2 \circ \Phi_1 + f(\mathbf{0}) -$
 562 $\omega_f(\sqrt{d})$, it holds that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Thus, we finish the proof.

563 3.3 Proof of Proposition 3.1

564 **Lemma 3.3.** *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*
 565 *$x_0 < x_1 < \dots < x_{N_1(N_2+1)}$ and $y_i \geq 0$ for $i = 0, 1, \dots, N_1(N_2+1)$, there exists $\phi \in \mathcal{NN}(\#input =$*
 566 *$1; \text{widthvec} = [2N_1, 2N_2 + 1]; \#output = 1)$ satisfying the following conditions.*

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

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

569 In fact, Lemma 3.3 is a part of Lemma 2.2 in [61]. For the purpose of being self-
570 contained, we present it as follows.

571 **Lemma** (Lemma 2.2 of [61]). For any $m, n \in \mathbb{N}^+$, given any $m(n+1)+1$ samples $(x_i, y_i) \in$
572 \mathbb{R}^2 with $x_0 < x_1 < x_2 < \dots < x_{m(n+1)}$ and $y_i \geq 0$ for $i = 0, 1, \dots, m(n+1)$, there exists
573 $\phi \in \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2m, 2n + 1]; \#\text{output} = 1)$ satisfying the following
574 conditions.

575 (i) $\phi(x_i) = y_i$ for $i = 0, 1, \dots, m(n+1)$;

576 (ii) ϕ is linear on each interval $[x_{i-1}, x_i]$ for $i \notin \{(n+1)j : j = 1, 2, \dots, m\}$;

577 (iii) $\sup_{x \in [x_0, x_{m(n+1)}]} |\phi(x)| \leq 3 \max_{i \in \{0, 1, \dots, m(n+1)\}} y_i \prod_{k=1}^n \left(1 + \frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} \right)$.

578 **Lemma 3.4.** Given any $N, L, d \in \mathbb{N}^+$, it holds that

$$\begin{aligned} & \mathcal{NN}(\#\text{input} = d; \text{widthvec} = [N, NL]; \#\text{output} = 1) \\ 579 & \subseteq \mathcal{NN}(\#\text{input} = d; \text{width} \leq 2N + 2; \text{depth} \leq L + 1; \#\text{output} = 1). \end{aligned}$$

580 *Proof.* The key idea to prove Proposition 3.4 is to re-assemble $\mathcal{O}(L)$ sub-FNNs in the
581 shallower FNN in the left of Figure 8 to form a deeper one with width $\mathcal{O}(N)$ and depth
 $\mathcal{O}(L)$ on the right of Figure 8.

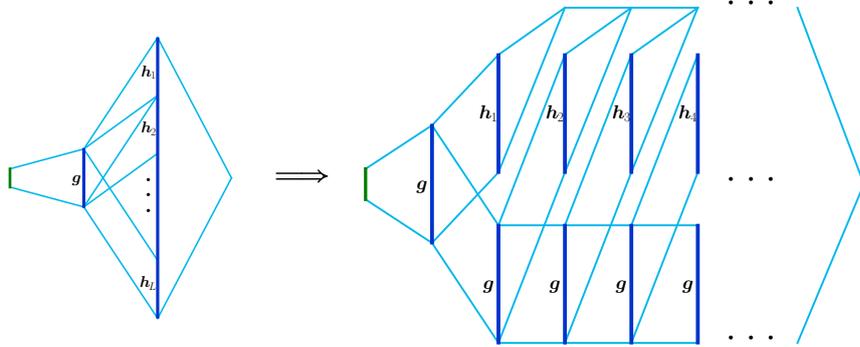


Figure 8: An illustration of the main idea to prove Lemma 3.4.

582

583 For any $\phi \in \mathcal{NN}(\#\text{input} = d; \text{widthvec} = [N, NL]; \#\text{output} = 1)$, ϕ can be imple-
584 mented by a ReLU FNN described as

$$585 \quad \mathbf{x} \xrightarrow[\sigma]{\mathbf{W}_0, \mathbf{b}_0} \mathbf{g} \xrightarrow[\sigma]{\mathbf{W}_1, \mathbf{b}_1} \mathbf{h} \xrightarrow{\mathbf{W}_2, \mathbf{b}_2} \phi(\mathbf{x}),$$

586 where \mathbf{g} and \mathbf{h} are the output of the first hidden layer and the second hidden layer,
587 respectively. Note that

$$588 \quad \mathbf{g} = \sigma(\mathbf{W}_0 \cdot \mathbf{x} + \mathbf{b}_0), \quad \mathbf{h} = \sigma(\mathbf{W}_1 \cdot \mathbf{g} + \mathbf{b}_1), \quad \text{and} \quad \phi(\mathbf{x}) = \mathbf{W}_2 \cdot \mathbf{h} + \mathbf{b}_2.$$

589 We can evenly divide $\mathbf{h} \in \mathbb{R}^{NL \times 1}$, $\mathbf{b}_1 \in \mathbb{R}^{NL \times 1}$, $\mathbf{W}_1 \in \mathbb{R}^{NL \times N}$, and $\mathbf{W}_2 \in \mathbb{R}^{1 \times NL}$ into L parts
590 as follows:

$$591 \quad \mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_L \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} \mathbf{b}_{1,1} \\ \mathbf{b}_{1,2} \\ \vdots \\ \mathbf{b}_{1,L} \end{bmatrix}, \quad \mathbf{W}_1 = \begin{bmatrix} \mathbf{W}_{1,1} \\ \mathbf{W}_{1,2} \\ \vdots \\ \mathbf{W}_{1,L} \end{bmatrix},$$

592 and $\mathbf{W}_2 = [\mathbf{W}_{2,1}, \mathbf{W}_{2,2}, \dots, \mathbf{W}_{2,L}]$, where $\mathbf{h}_\ell \in \mathbb{R}^{N \times 1}$, $\mathbf{b}_{1,\ell} \in \mathbb{R}^{N \times 1}$, $\mathbf{W}_{1,\ell} \in \mathbb{R}^{N \times N}$, and $\mathbf{W}_{2,\ell} \in$
593 $\mathbb{R}^{1 \times N}$ for $\ell = 1, 2, \dots, L$. Then, for $\ell = 1, 2, \dots, L$, we have

$$594 \quad \mathbf{h}_\ell = \sigma(\mathbf{W}_{1,\ell} \cdot \mathbf{g} + \mathbf{b}_{1,\ell}) \quad \text{and} \quad \phi(\mathbf{x}) = \mathbf{W}_2 \cdot \mathbf{h} + \mathbf{b}_2 = \sum_{j=1}^L \mathbf{W}_{2,j} \cdot \mathbf{h}_j + \mathbf{b}_2. \quad (3.7)$$

595

596 Define

$$597 \quad s_0 := 0, \quad \text{and} \quad s_\ell := \sum_{j=1}^{\ell} \mathbf{W}_{2,j} \cdot \mathbf{h}_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

598 Then $\phi(\mathbf{x}) = \mathbf{W}_2 \cdot \mathbf{h} + \mathbf{b}_2 = s_L + \mathbf{b}_2$ and

$$599 \quad s_\ell = s_{\ell-1} + \mathbf{W}_{2,\ell} \cdot \mathbf{h}_\ell, \quad \text{for } \ell = 1, 2, \dots, L. \quad (3.8)$$

600 Hence, it is easy to check that ϕ can also be implemented by the deep network shown
in Figure 9. It is clear that the network has the architecture of Figure 9 is with width

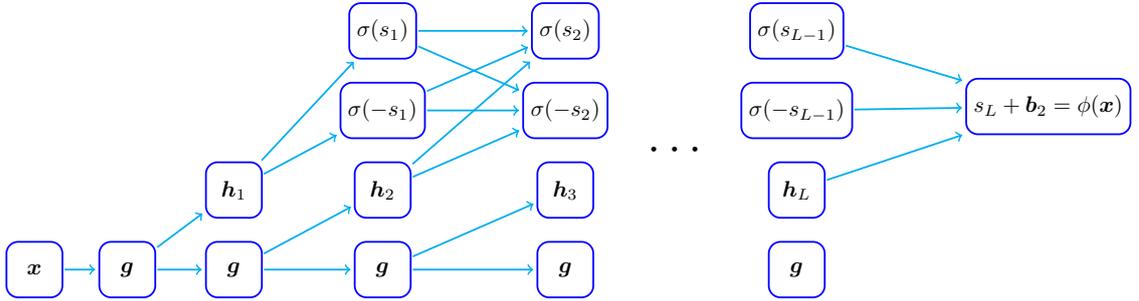


Figure 9: A illustration of the desired network based on Equation (3.7) and (3.8), and the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$. We omit the activation function (σ) if the input is non-negative.

601

602 $2N + 2$ and depth $L + 1$. So, we finish the proof. \square

603 With Lemma 3.3 and 3.4 in hand, we are ready to present the detailed proof of
604 Proposition 3.1.

605 *Proof of Proposition 3.1.* We divide the proof into two cases: $d = 1$ and $d \geq 2$.

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

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

$$609 \quad \{(1, M - 1), (2, 0)\} \cup \left\{ \left(\frac{m}{M}, m \right) : m = 0, 1, \dots, M - 1 \right\} \cup \left\{ \left(\frac{m+1}{M} - \delta, m \right) : m = 0, 1, \dots, M - 2 \right\}.$$

610 Its size is $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = N$ and $N_2 = 2NL - 1$
611 therein), there exists $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL - 1) + 1]) = \mathcal{NN}(\text{widthvec} =$
612 $[2N, 4NL - 1])$ such that

- 613 • $\phi_1(\frac{M-1}{M}) = \phi_1(1) = M - 1$ and $\phi_1(\frac{m}{M}) = \phi_1(\frac{m+1}{M} - \delta) = m$ for $m = 0, 1, \dots, M - 2$;
- 614 • ϕ_1 is linear on $[\frac{M-1}{M}, 1]$ and each interval $[\frac{m}{M}, \frac{m+1}{M} - \delta]$ for $m = 0, 1, \dots, M - 2$.

615 Then

$$616 \quad \phi_1(x) = m, \quad \text{if } x \in [\frac{m}{M}, \frac{m+1}{M} - \delta \cdot 1_{\{m \leq M-2\}}], \quad \text{for } m = 0, 1, \dots, M - 1. \quad (3.9)$$

617 Now consider the another sample set

$$618 \quad \{(\frac{1}{M}, L - 1), (2, 0)\} \cup \{(\frac{\ell}{ML}, \ell) : \ell = 0, 1, \dots, L - 1\} \cup \{(\frac{\ell+1}{ML} - \delta, \ell) : \ell = 0, 1, \dots, L - 2\}.$$

619 Its size is $2L + 1 = 1 \cdot ((2L - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = 1$ and $N_2 = 2L - 1$ therein),
620 there exists $\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 2(2L - 1) + 1]) = \mathcal{NN}(\text{widthvec} = [2, 4L - 1])$ such
621 that

- 622 • $\phi_2(\frac{L-1}{ML}) = \phi_2(\frac{1}{M}) = L - 1$ and $\phi_2(\frac{\ell}{ML}) = \phi_2(\frac{\ell+1}{ML} - \delta) = \ell$ for $\ell = 0, 1, \dots, L - 2$;
- 623 • ϕ_2 is linear on $[\frac{L-1}{ML}, \frac{1}{M}]$ and each interval $[\frac{\ell}{ML}, \frac{\ell+1}{ML} - \delta]$ for $\ell = 0, 1, \dots, L - 2$.

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

$$625 \quad \phi_2(x - \frac{m}{M}) = \ell, \quad \text{for } x \in [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{\ell \leq L-2\}}]. \quad (3.10)$$

626 The fact $K = ML$ implies each $k \in \{0, 1, \dots, K - 1\}$ can be unique represented by
627 $k = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. Then the desired function ϕ can
628 be implemented by a ReLU FNN shown in Figure 10. Clearly,

$$629 \quad \phi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}] \text{ for } k \in \{0, 1, \dots, K - 1\}.$$

630 By Lemma 3.4, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \subseteq \mathcal{NN}(\text{width} \leq 4N + 2; \text{depth} \leq 2L + 1)$
631 and $\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 4L - 1]) \subseteq \mathcal{NN}(\text{width} \leq 6; \text{depth} \leq 2L + 1)$, implying
632 $\phi \in \mathcal{NN}(\text{width} \leq \max\{4N + 2 + 1, 6 + 1\} = 4N + 3; \text{depth} \leq (2L + 1) + 2 + (2L + 1) + 1 = 4L + 5)$.
So we finish the proof for the case $d = 1$.

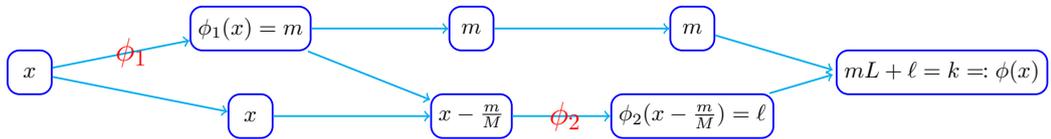


Figure 10: An illustration of the ReLU FNN implementing ϕ based on Equation (3.9) and (3.10) with $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}] = [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{m \leq M-2 \text{ or } \ell \leq L-2\}}]$, where $k = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

633

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

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

$$636 \quad \{(1, K-1), (2, 0)\} \cup \left\{ \left(\frac{k}{K}, k \right) : k = 0, 1, \dots, K-1 \right\} \cup \left\{ \left(\frac{k+1}{K} - \delta, k \right) : k = 0, 1, \dots, K-2 \right\},$$

637 whose size is $2K+1 = \lfloor N^{1/d} \rfloor ((2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = \lfloor N^{1/d} \rfloor$
638 and $N_2 = 2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1$ therein), there exists ϕ in

$$639 \quad \begin{aligned} & \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 2(2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1]) \\ & \subseteq \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \end{aligned}$$

640 such that

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

643 Then

$$644 \quad \phi(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}} \right] \text{ for } k = 0, 1, \dots, K-1.$$

645 By Lemma 3.4,

$$646 \quad \begin{aligned} & \phi \in \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \\ & \subseteq \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 2; \text{depth} \leq 2\lfloor L^{2/d} \rfloor + 1) \\ & \subseteq \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 4L + 5). \end{aligned}$$

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

648 **3.4 Proof of Proposition 3.2**

649 The proof of Proposition 3.2 is based on the bit extraction technique in [5, 27]. In
650 fact, we modify this technique to extract the sum of many bits rather than one bit and
651 this modification can be summarized in Lemma 3.5 and 3.6 below.

652 **Lemma 3.5.** *For any $L \in \mathbb{N}^+$, there exists a function ϕ in*

$$653 \quad \mathcal{NN}(\#\text{input} = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1; \#\text{output} = 1)$$

654 such that, for any $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$, we have

$$655 \quad \phi(\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell) = \sum_{j=1}^{\ell} \theta_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

656 *Proof.* Given $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$, define

$$657 \quad \xi_j := \text{bin}0.\theta_j\theta_{j+1}\cdots\theta_L, \quad \text{for } j = 1, 2, \dots, L$$

658 and

$$659 \quad \mathcal{T}(x) := \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

660 Then we have

$$661 \quad \theta_j = \mathcal{T}(\xi_j - 1/2), \quad \text{for } j = 1, 2, \dots, L,$$

662 and

$$663 \quad \xi_{j+1} = 2\xi_j - \theta_j, \quad \text{for } j = 1, 2, \dots, L - 1.$$

664 We would like to point out that, by above two iteration equations, we can iteratively get
 665 $\xi_1, \theta_1, \xi_2, \theta_2, \dots, \xi_L, \theta_L$ when ξ_1 is given. Based on this iteration idea, the rest proof can be
 666 divided into three steps.

667 **Step 1:** Simplify the iteration equations.

668 Note that $\mathcal{T}(x) = \sigma(x/\delta + 1) - \sigma(x/\delta)$ for any $x \notin (-\delta, 0)$. By setting $\delta = 1/2 - \sum_{j=2}^L 2^{-j} =$
 669 2^{-L} , we have $\xi_j - 1/2 \notin (-\delta, 0)$ for all j , implying

$$670 \quad \begin{aligned} \theta_j = \mathcal{T}(\xi_j - 1/2) &= \sigma((\xi_j - 1/2)/\delta + 1) - \sigma((\xi_j - 1/2)/\delta) \\ &= \sigma(\mathcal{L}(\xi_j) + 1) - \sigma(\mathcal{L}(\xi_j)), \end{aligned} \quad (3.11)$$

671 for $j = 1, 2, \dots, L$, where \mathcal{L} is the linear map given by $\mathcal{L}(x) = (x - 1/2)/\delta$. It follows that,
 672 for $j = 1, 2, \dots, L - 1$,

$$673 \quad \xi_{j+1} = 2\xi_j - \theta_j = 2\xi_j - \sigma(\mathcal{L}(\xi_j) + 1) + \sigma(\mathcal{L}(\xi_j)). \quad (3.12)$$

674 **Step 2:** Design a ReLU FNN to output $\sum_{j=1}^{\ell} \theta_j$.

675 It is easy to design a ReLU FNN to output $\theta_1, \theta_2, \dots, \theta_L$ by Equation (3.11) and
 676 (3.12) when using $\xi_1 = \text{bin}0.\theta_1\theta_2\dots\theta_L$ as the input. However, it is highly non-trivial to
 677 construct a ReLU FNN to output $\sum_{j=1}^{\ell} \theta_j$ with another input ℓ , since many operations
 678 like multiplication and comparison are not allowed in designing ReLU FNNs.

679 Now let us establish a formula to represent $\sum_{j=1}^{\ell} \theta_j$ in a form of a ReLU FNN as
 680 follows:

681 The fact that $x_1x_2 = \sigma(x_1 + x_2 - 1)$ for any $x_1, x_2 \in \{0, 1\}$ implies

$$682 \quad \begin{aligned} \sum_{j=1}^{\ell} \theta_j &= \sum_{j=1}^L \theta_j \mathcal{T}(\ell - j) = \sum_{j=1}^L \sigma(\theta_j + \mathcal{T}(\ell - j) - 1) \\ &= \sum_{j=1}^L \sigma(\theta_j + \sigma(\ell - j + 1) - \sigma(\ell - j) - 1), \end{aligned}$$

683 for $\ell = 1, 2, \dots, L$, where the last equality comes from the fact $\mathcal{T}(n) = \sigma(n + 1) - \sigma(n)$ for
 684 any integer n .

685 To simplify the notations, we define

$$686 \quad z_{\ell,j} := \sigma(\theta_j + \sigma(\ell - j + 1) - \sigma(\ell - j) - 1), \quad (3.13)$$

687 for $\ell = 1, 2, \dots, L$ and $j = 1, 2, \dots, L$. Then,

$$688 \quad \sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^L z_{\ell,j}, \quad \text{for } \ell = 1, 2, \dots, L. \quad (3.14)$$

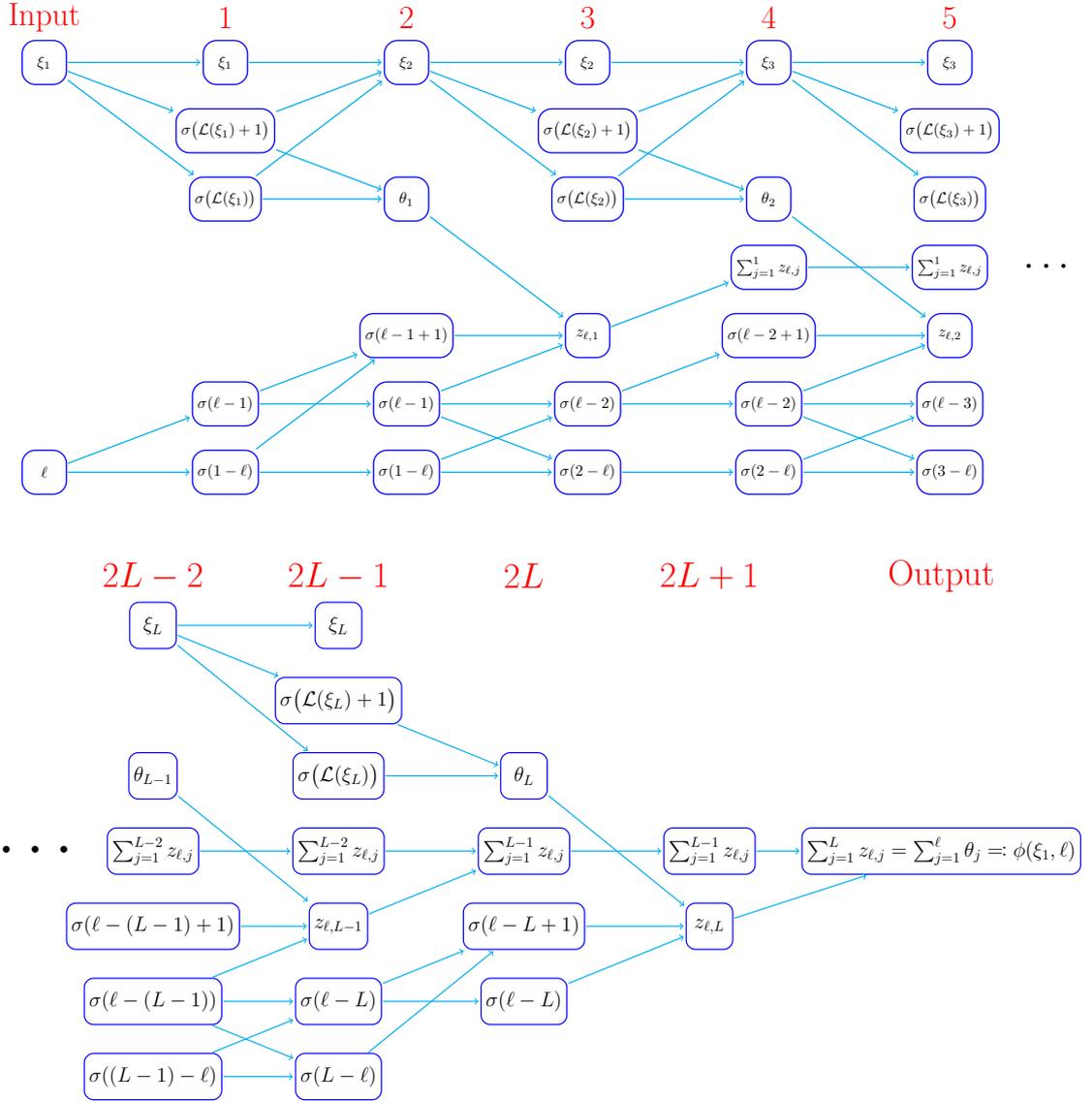


Figure 11: A illustration of the target ReLU FNN implementing ϕ to output $\sum_{j=1}^L z_{j,\ell} = \sum_{j=1}^L \theta_j = \phi(\xi_1, \ell)$ given the input $(\xi_1, \ell) = (\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \dots, L\}$ and $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$. The construction is mainly based on Equation (3.11), (3.12), (3.13), and (3.14). The numbers above the architecture indicate the order of hidden layers. It builds a whole iteration step for every two layers. We output both $\sigma(\ell - j)$ and $\sigma(j - \ell)$ in the hidden layers for $j = 1, 2, \dots, L$ because of the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$. We omit the activation function (σ) if the input of a neuron is non-negative. Note that all parameters of this network are essentially determined by Equation (3.11) and (3.12), which are valid no matter what $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$ are. Thus, the desired function ϕ implemented by this network is independent of $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$.

689 With Equation (3.11), (3.12), (3.13), and (3.14) in hand, it is easy to construct a
690 function ϕ implemented by a ReLU FNN with the desired width and depth outputting
691 $\sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^L z_{\ell,j}$ given the input $(\xi_1, \ell) = (\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \dots, L\}$ and
692 $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$. The details of construction are shown in Figure 11. Clearly, the
693 network in Figure 11 is with width 7 and depth $2L + 1$, which implies

$$694 \quad \phi \in \mathcal{NN}(\#\text{input} = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1; \#\text{output} = 1).$$

695 So we finish the proof. \square

696 Next, we introduce Lemma 3.6 as an advanced version of Lemma 3.5.

697 **Lemma 3.6.** *For any $N, L \in \mathbb{N}^+$, any $\theta_{m,\ell} \in \{0, 1\}$ for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-$
698 1 , where $M = N^2L$, there exists a function ϕ implemented by a ReLU FNN with width
699 $4N + 3$ and depth $3L + 3$ such that*

$$700 \quad \phi(m, \ell) = \sum_{j=0}^{\ell} \theta_{m,j}, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1.$$

701 *Proof.* Define

$$702 \quad y_m := \text{bin}0.\theta_{m,0}\theta_{m,1}\cdots\theta_{m,L-1}, \quad \text{for } m = 0, 1, \dots, M-1.$$

703 Consider the sample set $\{(m, y_m) : m = 0, 1, \dots, M\}$, whose cardinality is $M+1 = N((NL-$
704 $1) + 1) + 1$. By Lemma 3.3 (set $N_1 = N$ and $N_2 = NL - 1$ therein), there exists

$$705 \quad \begin{aligned} \phi_1 &\in \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2(NL - 1) + 1]) \\ &= \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2NL - 1]) \end{aligned}$$

706 such that

$$707 \quad \phi_1(m) = y_m, \quad \text{for } m = 0, 1, \dots, M-1.$$

708 By Lemma 3.5, there exists

$$709 \quad \phi_2 \in \mathcal{NN}(\#\text{input} = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1)$$

710 such that, for any $\xi_1, \xi_2, \dots, \xi_L \in \{0, 1\}$, we have

$$711 \quad \phi_2(\text{bin}0.\xi_1\xi_2\cdots\xi_L, \ell) = \sum_{j=1}^{\ell} \xi_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

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

$$713 \quad \phi_2(\text{bin}0.\xi_0\xi_1\cdots\xi_{L-1}, \ell + 1) = \sum_{j=0}^{\ell} \xi_j, \quad \text{for } \ell = 0, 1, \dots, L-1.$$

714 Thus, for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$, we have

$$715 \quad \phi_2(\phi_1(m), \ell + 1) = \phi_2(y_m, \ell + 1) = \phi_2(0.\theta_{m,0}\theta_{m,1}\cdots\theta_{m,L-1}, \ell + 1) = \sum_{j=0}^{\ell} \theta_{m,j}.$$

716 Hence, the desired function ϕ can be implemented by the network shown
717 in Figure 12. By Lemma 3.4, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL - 1]) \subseteq \mathcal{NN}(\text{width} \leq 4N +$
718 $2; \text{depth} \leq L + 1)$, implying the network in Figure 12 is with width $\max\{(4N + 2) + 1, 7\} =$
719 $4N + 3$ and depth $(2L + 1) + 1 + (L + 1) = 3L + 3$. So we finish the proof. \square

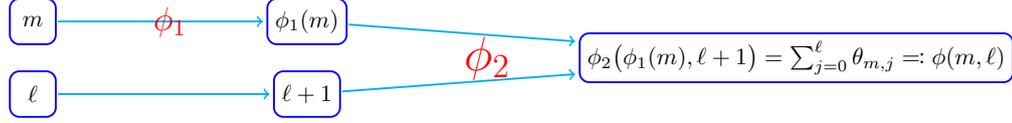


Figure 12: A illustration of the network implementing the desired function ϕ . “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

720 Next, we apply Lemma 3.6 to prove Lemma 3.7 below, which is a key intermediate
721 conclusion to prove Proposition 3.2.

722 **Lemma 3.7.** For any $\varepsilon > 0$, $L, N \in \mathbb{N}^+$, denote $M = N^2L$ and assume $\{y_{m,\ell} \geq 0 : m =$
723 $0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1\}$ is a sample set with

$$724 \quad |y_{m,\ell} - y_{m,\ell-1}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 1, 2, \dots, L-1.$$

725 Then there exists $\phi \in \mathcal{NN}$ (#input = 2; width $\leq 12N + 8$; depth $\leq 3L + 6$) such that

726 (i) $|\phi(m, \ell) - y_{m,\ell}| \leq \varepsilon$, for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$;

727 (ii) $0 \leq \phi(x_1, x_2) \leq \max\{y_{m,\ell} : m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1\}$, for any $x_1, x_2 \in \mathbb{R}$.

728 *Proof.* Define

$$729 \quad a_{m,\ell} := \lfloor y_{m,\ell}/\varepsilon \rfloor, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1.$$

730 We will construct a function implemented by a ReLU FNN to map the index (m, ℓ) to
731 $a_{m,\ell}\varepsilon$ for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$.

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

$$735 \quad \begin{aligned} a_{m,\ell} &= a_{m,0} + \sum_{j=1}^{\ell} (a_{m,j} - a_{m,j-1}) = a_{m,0} + \sum_{j=1}^{\ell} b_{m,j} = a_{m,0} + \sum_{j=0}^{\ell} b_{m,j} \\ &= a_{m,0} + \sum_{j=0}^{\ell} c_{m,j} - \sum_{j=0}^{\ell} d_{m,j}. \end{aligned}$$

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

737 For the sample set $\{(m, a_{m,0}) : m = 0, 1, \dots, M-1\} \cup \{(M, 0)\}$, whose size is $M+1 =$
738 $N \cdot ((NL-1) + 1) + 1$, by Lemma 3.3 (set $N_1 = N$ and $N_2 = NL-1$ therein), there exists
739 $\psi_1 \in \mathcal{NN}$ (widthvec = $[2N, 2(NL-1) + 1]$) = \mathcal{NN} (widthvec = $[2N, 2NL-1]$) such that

$$740 \quad \psi_1(m) = a_{m,0}, \quad \text{for } m = 0, 1, \dots, M-1.$$

741 By Lemma 3.6, there exist $\psi_2, \psi_3 \in \mathcal{NN}$ (width $\leq 4N + 3$; depth $\leq 3L + 3$) such that

$$742 \quad \psi_2(m, \ell) = \sum_{j=0}^{\ell} c_{m,j} \quad \text{and} \quad \psi_3(m, \ell) = \sum_{j=0}^{\ell} d_{m,j},$$

743 for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. Hence, it holds that

$$744 \quad a_{m,\ell} = a_{m,0} + \sum_{j=0}^{\ell} c_{m,j} - \sum_{j=0}^{\ell} d_{m,j} = \psi_1(m) + \psi_2(m, \ell) - \psi_3(m, \ell), \quad (3.15)$$

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

746 Define

$$747 \quad y_{\max} := \max\{y_{m,\ell} : m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1\}.$$

748 Then the desired function can be implemented by two sub-networks shown in Figure 13.

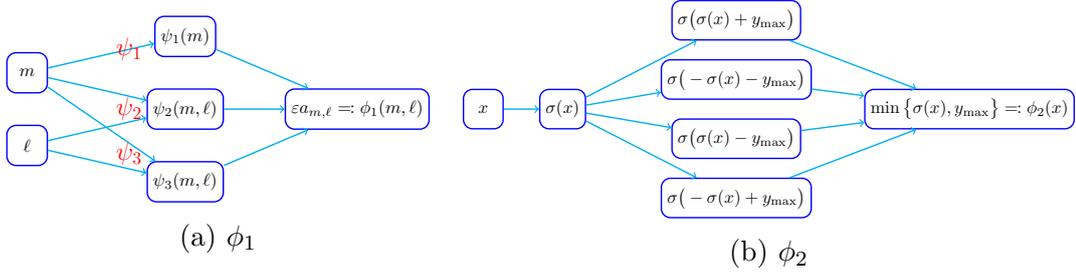


Figure 13: Illustrations of two sub-networks implementing the desired function $\phi = \phi_2 \circ \phi_1$ based Equation (3.15) and the fact $\min\{x_1, x_2\} = \frac{x_1 + x_2 - |x_1 - x_2|}{2} = \frac{\sigma(x_1 + x_2) - \sigma(-x_1 - x_2) - \sigma(x_1 - x_2) - \sigma(-x_1 + x_2)}{2}$. y_{\max} is given by $\max\{y_{m,\ell} : m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1\}$. “ ψ_1 ”, “ ψ_2 ”, and “ ψ_3 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

749 By Lemma 3.4, $\psi_1 \in \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2NL - 1]) \subseteq \mathcal{NN}(\#\text{input} =$
750 $1; \text{width} \leq 4N + 2; \text{depth} \leq L + 1)$. Note that $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 4N + 3; \text{depth} \leq 3L + 3)$.
751 Thus, $\phi_1 \in \mathcal{NN}(\text{width} \leq (4N + 2) + 2(4N + 3) = 12N + 8; \text{depth} \leq (3L + 3) + 1 = 3L + 4)$
752 as shown in Figure 13. And it is clear that $\phi_2 \in \mathcal{NN}(\text{width} \leq 4; \text{depth} \leq 2)$, implying
753 $\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq (3L + 4) + 2 = 3L + 6)$.

754 Clearly, $0 \leq \phi(x_1, x_2) \leq y_{\max}$ for any $x_1, x_2 \in \mathbb{R}$, since $\phi(x_1, x_2) = \phi_2 \circ \phi_1(x_1, x_2) =$
755 $\max\{\sigma(\phi_1(x_1, x_2)), y_{\max}\}$.

756 Note that $0 \leq \varepsilon a_{m,\ell} = \varepsilon \lfloor y_{m,\ell} / \varepsilon \rfloor \leq y_{\max}$. Then we have $\phi(m, \ell) = \phi_2 \circ \phi_1(m, \ell) =$
757 $\phi_2(\varepsilon a_{m,\ell}) = \max\{\sigma(\varepsilon a_{m,\ell}), y_{\max}\} = \varepsilon a_{m,\ell}$. Therefore,

$$758 \quad |\phi(m, \ell) - y_{m,\ell}| = |a_{m,\ell} \varepsilon - y_{m,\ell}| = \left| \lfloor y_{m,\ell} / \varepsilon \rfloor \varepsilon - y_{m,\ell} \right| \leq \varepsilon,$$

759 for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. Hence, we finish the proof. \square

760 Finally, we apply Lemma 3.7 to prove Proposition 3.2.

761 *Proof of Proposition 3.2.* Let $M = N^2L$, then we may assume $J = ML$ since we can set
762 $y_{J-1} = y_J = y_{J+1} = \dots = y_{ML-1}$ if $J < ML$.

763 For the sample set

$$764 \quad \{(mL, m) : m = 0, 1, \dots, M\} \cup \{(mL + L - 1, m) : m = 0, 1, \dots, M - 1\},$$

765 whose size is $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$, by Lemma 3.3 (set $N_1 = N$ and $N_2 =$
766 $NL - 1$ therein), there exist $\phi_1 \in \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2(2NL - 1) + 1]) =$
767 $\mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 4NL - 1])$ such that

- 768 • $\phi_1(ML) = M$ and $\phi_1(mL) = \phi_1(mL + L - 1) = m$ for $m = 0, 1, \dots, M - 1$;
- 769 • ϕ_1 is linear on each interval $[mL, mL + L - 1]$ for $m = 0, 1, \dots, M - 1$.

770 It follows that

$$771 \quad \phi_1(j) = m, \quad \text{and} \quad j - L\phi_1(j) = \ell, \quad \text{where } j = mL + \ell, \quad (3.16)$$

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

773 Note that any number j in $\{0, 1, \dots, J - 1\}$ can be uniquely indexed as $j = mL + \ell$
774 for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. So we can denote $y_j = y_{mL + \ell}$ as $y_{m, \ell}$. Then by
775 Lemma 3.7, there exists $\phi_2 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq 3L + 6)$ such that

$$776 \quad |\phi_2(m, \ell) - y_{m, \ell}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1, \quad (3.17)$$

777 and

$$778 \quad 0 \leq \phi_2(x_1, x_2) \leq y_{\max}, \quad \text{for any } x_1, x_2 \in \mathbb{R}, \quad (3.18)$$

779 where $y_{\max} := \max\{y_{m, \ell} : m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1\} = \max\{y_j : j =$
780 $0, 1, \dots, ML - 1\}$.

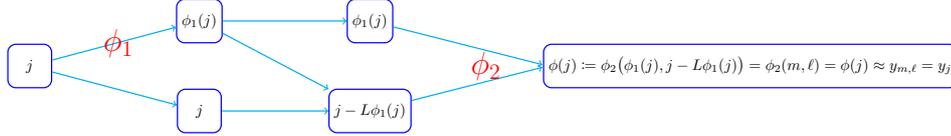


Figure 14: A illustration of the ReLU FNN implementing the desired function ϕ based Equation (3.16). The index $j \in \{0, 1, \dots, ML - 1\}$ is unique represented by $j = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

781 Note that $\phi_1 \in \mathcal{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 4NL - 1]) \subseteq \mathcal{NN}(\#\text{input} =$
782 $1; \text{width} \leq 8N + 2; \text{depth} \leq L + 1)$ by Lemma 3.4 and $\phi_2 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq$
783 $3L + 6)$. So $\phi \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq (L + 1) + 2 + (3L + 6) = 4L + 9)$ as shown in
784 Figure 14.

785 Equation (3.18) implies

$$786 \quad 0 \leq \phi(x) \leq y_{\max}, \quad \text{for any } x \in \mathbb{R},$$

787 since ϕ is given by $\phi(x) = \phi_2(\phi_1(x), x - L\phi_1(x))$.

788 Represent $j \in \{0, 1, \dots, ML - 1\}$ via $j = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$,
789 then we have, by Equation (3.17),

$$790 \quad |\phi(j) - y_j| = |\phi_2(\phi_1(j), j - L\phi_1(j)) - y_j| = |\phi_2(m, \ell) - y_{m, \ell}| \leq \varepsilon.$$

791 So we finish the proof. □

792 We would like to remark that the key idea in the proof of Proposition 3.2 is the bit
 793 extraction technique in Lemma 3.5, which allows us to store L bits in a binary number
 794 $\text{bin}0.\theta_1\theta_2\cdots\theta_L$ and extract each bit θ_i . The extraction operator can be efficiently carried
 795 out via a deep ReLU neural network demonstrating the power of depth.

796 4 Neural networks approximation and evaluation in 797 practice

798 This section is concerned with neural networks approximation and evaluation in
 799 practice, e.g., approximating functions defined on irregular domains or domains with a
 800 low-dimensional structure, and neural network computation in parallel computing. In
 801 the practical training of FNNs, the approximation rate in this paper can only be observed
 802 if the global minimizers of neural network optimization can be identified. Since there is
 803 no existing optimization algorithm guaranteeing a global minimizer, it is challenging to
 804 observe the proved approximation rate currently. Developing optimization algorithms
 805 for global minimizers is another interesting research topic as a future work.

806 4.1 Approximation on irregular domain

807 In this section, we consider approximating continuous functions defined on irregular
 808 domains by deep ReLU FNNs. The construction is through extending the target function
 809 to a cubic domain, applying Theorem 1.1, and finally restricting the constructed FNN
 810 back to the irregular domain.

811 Given any uniformly continuous and real-valued function f defined on a metric space
 812 S with a metric $d_S(\cdot, \cdot)$, we define the (optimal) modulus of continuity of f on a subset
 813 $E \subseteq S$ as

$$814 \quad \omega_f^E(r) := \sup\{|f(\mathbf{x}_1) - f(\mathbf{x}_2)| : d_S(\mathbf{x}_1, \mathbf{x}_2) \leq r, \mathbf{x}_1, \mathbf{x}_2 \in E\}, \quad \text{for any } r \geq 0.$$

815 For the purpose of consistency and simplicity, $\omega_f(\cdot)$ is short of $\omega_f^{[0,1]^d}(\cdot)$.

816 First, let us present two lemmas for (approximately) extending (almost) continuous
 817 functions on E to (almost) continuous functions on S . These lemmas are similar to
 818 the well-known results for extending Lipschitz or differentiable functions in [46, 63]. We
 819 generalize these results to a broader class of functions required in the proof of Theorem
 820 4.3.

821 **Lemma 4.1** (Approximate Extension of Almost-Continuous Functions). *Assume S is a*
 822 *metric space with a metric $d_S(\cdot, \cdot)$ and $\omega : [0, \infty) \rightarrow [0, \infty)$ is an increasing function with*

$$823 \quad \omega(r_1 + r_2) \leq \omega(r_1) + \omega(r_2), \quad \text{for any } r_1, r_2 \in [0, \infty). \quad (4.1)$$

824 *Let f be a real-valued function defined on a subset $E \subseteq S$ and satisfy*

$$825 \quad |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2) + \Delta), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in E, \quad (4.2)$$

826 where Δ is a positive constant independent of f . Then there exists a function g defined
 827 on S such that

$$828 \quad 0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega(\Delta), \quad \text{for any } \mathbf{x} \in E$$

829 and

$$830 \quad |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in S.$$

831 In Lemma 4.1, g is an approximate extension of f defined on E to a new domain S
 832 with an approximation error $\omega(\Delta)$. In a special case when $\Delta = 0$ and $\omega(0) = 0$, g is an
 833 exact extension of f .

834 *Proof of Lemma 4.1.* Define

$$835 \quad g(\mathbf{x}) := \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}) + \Delta)).$$

836 By Equation (4.2), we have $f(\mathbf{x}_1) - \omega(d_S(\mathbf{x}_1, \mathbf{x}_2) + \Delta) \leq f(\mathbf{x}_2)$ for any $\mathbf{x}_1, \mathbf{x}_2 \in E$. It
 837 holds that $g(\mathbf{x}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in E$. Together with

$$838 \quad g(\mathbf{x}) = \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}) + \Delta)) \geq f(\mathbf{x}) - \omega(d_S(\mathbf{x}, \mathbf{x}) + \Delta) = f(\mathbf{x}) - \omega(\Delta),$$

839 for any $\mathbf{x} \in E$, it follows that $0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega(\Delta)$ for any $\mathbf{x} \in E$. By Equation (4.1)
 840 and the fact

$$841 \quad \sup_{\mathbf{z} \in E} f_1(\mathbf{z}) - \sup_{\mathbf{z} \in E} f_2(\mathbf{z}) \leq \sup_{\mathbf{z} \in E} (f_1(\mathbf{z}) - f_2(\mathbf{z})), \quad \text{for any functions } f_1, f_2,$$

842 we have

$$\begin{aligned} 843 \quad g(\mathbf{x}_1) - g(\mathbf{x}_2) &= \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}_1))) - \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}_2))) \\ &\leq \sup_{\mathbf{z} \in E} (\omega(d_S(\mathbf{z}, \mathbf{x}_1)) - \omega(d_S(\mathbf{z}, \mathbf{x}_2))) \\ &\leq \sup_{\mathbf{z} \in E} \omega(d_S(\mathbf{z}, \mathbf{x}_1) - d_S(\mathbf{z}, \mathbf{x}_2)) \\ &\leq \sup_{\mathbf{z} \in E} \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)) = \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)), \end{aligned}$$

844 for any $\mathbf{x}_1, \mathbf{x}_2 \in S$. Similarly, we have $g(\mathbf{x}_2) - g(\mathbf{x}_1) \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2))$, which implies

$$845 \quad |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)).$$

846 So we finish the proof. □

847 Next, we introduce a lemma below for extending continuous functions defined on
 848 $E \subseteq S$ to continuous functions defined on S preserving the modulus of continuity.

849 **Lemma 4.2** (Extension of Continuous Functions). *Suppose f is a uniformly continuous*
 850 *function defined on a subset $E \subseteq S$, where S is a metric space with a metric $d_S(\cdot, \cdot)$, then*
 851 *there exists a uniformly continuous function g on S such that $f(\mathbf{x}) = g(\mathbf{x})$ for $\mathbf{x} \in E$ and*
 852 *$\omega_f^E(r) = \omega_g^S(r)$ for any $r \geq 0$.*

853 *Proof.* By the application of Lemma 4.1 with $\omega(r) = \omega_f^E(r)$ for $r \geq 0$ and $\Delta = 0$, we know
 854 that there exists $g : S \rightarrow \mathbb{R}$ such that

$$855 \quad 0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega_f^E(\Delta) = 0, \quad \text{for any } \mathbf{x} \in E,$$

856 and

$$857 \quad |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega_f^E(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in S.$$

858 The equation above and the uniform continuity of f imply that g is uniformly continuous.
 859 It also follows that

$$860 \quad f(\mathbf{x}) = g(\mathbf{x}), \quad \text{for any } \mathbf{x} \in E, \quad \text{and} \quad \omega_g^S(r) \leq \omega_f^E(r), \quad \text{for any } r \geq 0,$$

861 since $\omega_g^S(\cdot)$ is the optimal modulus of continuity of g . Note that $\omega_f^E(\cdot)$ is the optimal
 862 modulus of continuity of f and

$$863 \quad |f(\mathbf{x}_1) - f(\mathbf{x}_2)| = |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega_g^S(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in E.$$

864 Hence, $\omega_f^E(r) \leq \omega_g^S(r)$ for all $r \geq 0$, which implies $\omega_f^E(r) = \omega_g^S(r)$ since we have proved
 865 that $\omega_g^S(r) \leq \omega_f^E(r)$ for all $r \geq 0$. So we finish the proof. \square

866 Now we are ready to introduce and prove the main theorem of this section, which
 867 extends Theorem 1.1 to an irregular domain as follows.

868 **Theorem 4.3.** *Let f be a uniformly continuous function defined on $E \subseteq [-R, R]^d$. For*
 869 *arbitrary $L \in \mathbb{N}^+$ and $N \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU FNN*
 870 *with width $3^{d+3} \max\{d\lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+14+2d$ such that*

$$871 \quad \|f - \phi\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_f^E(2RN^{-2/d}L^{-2/d}).$$

872 *Proof.* By Lemma 4.2, f can be extended to \mathbb{R}^d such that

$$873 \quad \omega_f^{\mathbb{R}^d}(r) = \omega_f^E(r), \quad \text{for any } r \geq 0.$$

874 Define

$$875 \quad \tilde{f}(\mathbf{x}) := f(2R\mathbf{x} - R), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d.$$

876 It follows that

$$877 \quad \omega_{\tilde{f}}^{\mathbb{R}^d}(r) = \omega_f^{\mathbb{R}^d}(2Rr) = \omega_f^E(2Rr), \quad \text{for any } r \geq 0. \quad (4.3)$$

878 By Theorem 1.1, there exists a function $\tilde{\phi}$ implemented by a ReLU FNN with width
 879 $3^{d+3} \max\{d\lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+14+2d$ such that

$$880 \quad \|\tilde{f} - \tilde{\phi}\|_{L^\infty([0,1]^d)} \leq 19\sqrt{d}\omega_{\tilde{f}}^{[0,1]^d}(N^{-2/d}L^{-2/d}) \leq 19\sqrt{d}\omega_{\tilde{f}}^{\mathbb{R}^d}(N^{-2/d}L^{-2/d}).$$

881 Define

$$882 \quad \phi(\mathbf{x}) := \tilde{\phi}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d.$$

883 Then, by Equation (4.3), for any $\mathbf{x} \in E \subseteq [-R, R]^d$, we have

$$884 \quad \begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &= \left| \tilde{f}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right) - \tilde{\phi}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right) \right| \leq \|\tilde{f} - \tilde{\phi}\|_{L^\infty([0,1]^d)} \\ &\leq 19\sqrt{d}\omega_{\tilde{f}}^{\mathbb{R}^d}(N^{-2/d}L^{-2/d}) = 19\sqrt{d}\omega_f^E(2RN^{-2/d}L^{-2/d}), \end{aligned}$$

885 which implies

$$886 \quad \|f - \phi\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_f^E(2RN^{-2/d}L^{-2/d}).$$

887 So we finish the proof. \square

888 4.2 Approximation in a neighborhood of a low-dimensional man- 889 ifold

890 In this section, we study neural network approximation of functions defined in a
891 neighborhood of a low-dimensional manifold and prove Theorem 1.2 in this setting. Let
892 us first introduce Theorem 4.4 which is required to prove Theorem 1.2.

893 **Theorem 4.4** (Theorem 3.1 of [3]). *Let \mathcal{M} be a compact $d_{\mathcal{M}}$ -dimensional Riemannian*
894 *submanifold of \mathbb{R}^d having condition number $1/\tau$, volume V , and geodesic covering reg-*
895 *ularity \mathcal{R} . Fix $\delta \in (0, 1)$ and $\gamma \in (0, 1)$. Let $\mathbf{A} = \sqrt{\frac{d}{d_\delta}}\Phi$, where $\Phi \in \mathbb{R}^{d_\delta \times d}$ is a random*
896 *orthoprojector with*

$$897 \quad d_\delta = \mathcal{O}\left(\frac{d_{\mathcal{M}} \ln(dV\mathcal{R}\tau^{-1}\delta^{-1}) \ln(1/\gamma)}{\delta^2}\right).$$

898 *If $d_\delta \leq d$, then with probability at least $1 - \gamma$, the following statement holds: For every*
899 *$\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}$,*

$$900 \quad (1 - \delta)|\mathbf{x}_1 - \mathbf{x}_2| \leq |\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| \leq (1 + \delta)|\mathbf{x}_1 - \mathbf{x}_2|.$$

901 Theorem 4.4 shows the existence of a linear projector $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$ that maps a low-
902 dimensional manifold in a high-dimensional space to a low-dimensional space nearly
903 preserving distance. With this projection \mathbf{A} available, we can prove Theorem 1.2 via
904 constructing a ReLU FNN defined in the low-dimensional space using Theorem 4.3 and
905 hence the curse of dimensionality is lessened. The ideas of the proof are summarized in
906 the following Table 1.

907 In Table 1 and the detailed proof later, we introduce a new notation $\mathcal{SL}(E)$ for any
908 compact set $E \subseteq \mathbb{R}^d$ as the “smallest” element of E . Specifically, $\mathcal{SL}(E)$ is defined as the
909 unique point in $\cap_{k=1}^d E_k$, where

$$910 \quad E_k := \{\mathbf{x} \in E_{k-1} : x_k = s_k\}, \quad s_k := \inf\{x_k : [x_1, x_2, \dots, x_d]^T \in E_{k-1}\}, \quad \text{for } k = 1, 2, \dots, d,$$

911 and $E_0 = E$. The compactness of E ensures that $\cap_{k=1}^d E_k$ is in fact one point belong-
912 ing to E . The introduction of $\mathcal{SL}(\cdot)$ uniquely formulates a low-dimensional function \tilde{f}
913 representing a high-dimensional function f defined on \mathcal{M}_ε by

$$914 \quad \tilde{f}(\mathbf{y}) := f(\mathbf{x}_\mathbf{y}), \quad \text{where } \mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}), \quad \text{for any } \mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}.$$

915 As we shall see later, such a definition of \tilde{f} is reasonable because $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}$
916 is contained in a small ball of radius $\mathcal{O}(\varepsilon)$ for any $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$. There are many other
917 alternative ways to define $\mathcal{SL}(\cdot)$ as long as the definition ensures that $\mathcal{SL}(E)$ contains
918 only one element. For example, $\mathcal{SL}(E)$ can be defined as any arbitrary point in E . For
919 another example, $\mathbf{y} \in \mathbf{A}(\mathcal{M})$ cannot guarantee $\mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}) \in \mathcal{M}$ in the
920 current definition, but in practice we can choose $\mathcal{SL}(\{\mathbf{x} \in \mathcal{M} : \mathbf{A}\mathbf{x} = \mathbf{y}\})$ as $\mathbf{x}_\mathbf{y}$ to ensure
921 that $\mathbf{x}_\mathbf{y} \in \mathcal{M}$, which might be beneficial for potential applications.

922 Now we are ready to prove Theorem 1.2.

923 *Proof of Theorem 1.2.* By Theorem 4.4, there exists a matrix $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$ such that

$$924 \quad \mathbf{A}\mathbf{A}^T = \frac{d}{d_\delta} \mathbf{I}_{d_\delta}, \quad (4.4)$$

Table 1: Main steps of the proof of Theorem 1.2. Step 1: dimension reduction via the nearly isometric projection operator \mathbf{A} provided by Theorem 4.4 to obtain an “equivalent” function \tilde{f} of f in a low-dimensional domain using $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$. Step 2: construct a ReLU FNN to implement $\tilde{\phi} \approx \tilde{f}$ by Theorem 4.3. Step 3: define a ReLU FNN to implement ϕ in the original high-dimensional domain via the projection \mathbf{A} . Step 4: verify that the approximation error of $\phi \approx f$ satisfies our requirement.

$f(\mathbf{x})$ for $\mathbf{x} \in \mathcal{M}_\varepsilon \subseteq [0, 1]^d$	Step 4 \approx	$\phi(\mathbf{x}) := \tilde{\phi}(\mathbf{A}\mathbf{x})$ for $\mathbf{x} \in \mathcal{M}_\varepsilon \subseteq [0, 1]^d$
Step 1 \Downarrow $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$		Step 3 \Downarrow $\mathbf{y} = \mathbf{A}\mathbf{x}$
$\tilde{f}(\mathbf{y}) := f(\mathbf{x}_y)$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}$	Step 2 \approx	$\tilde{\phi}(\mathbf{y})$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}$

925 where \mathbf{I}_{d_δ} is an identity matrix of size $d_\delta \times d_\delta$, and

926
$$(1 - \delta)|\mathbf{x}_1 - \mathbf{x}_2| \leq |\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| \leq (1 + \delta)|\mathbf{x}_1 - \mathbf{x}_2|, \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}. \quad (4.5)$$

927 Given any $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$, then $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}$ is a nonzero compact set. Let
 928 $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$, then we define \tilde{f} on $\mathbf{A}(\mathcal{M}_\varepsilon)$ as $\tilde{f}(\mathbf{y}) = f(\mathbf{x}_y)$.

929 For any $\mathbf{y}_1, \mathbf{y}_2 \in \mathbf{A}(\mathcal{M}_\varepsilon)$, let $\mathbf{x}_i = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}_i\})$, then $\mathbf{x}_i \in \mathcal{M}_\varepsilon$ for $i = 1, 2$.
 930 By the definition of \mathcal{M}_ε , there exist $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{M}$ such that $|\tilde{\mathbf{x}}_i - \mathbf{x}_i| \leq \varepsilon$ for $i = 1, 2$. It
 931 follows that

932
$$|\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| = |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \omega_f(|\mathbf{x}_1 - \mathbf{x}_2|) \leq \omega_f(|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2| + 2\varepsilon) \leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\tilde{\mathbf{x}}_1 - \mathbf{A}\tilde{\mathbf{x}}_2| + 2\varepsilon\right),$$

933 where the last inequality comes from Equation (4.5). By the triangular inequality, we
 934 have

$$\begin{aligned} |\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| + \frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\tilde{\mathbf{x}}_1| + \frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_2 - \mathbf{A}\tilde{\mathbf{x}}_2| + 2\varepsilon\right) \\ 935 &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| + \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) \\ &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{y}_1 - \mathbf{y}_2| + \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right). \end{aligned}$$

936 Set $\omega(r) = \omega_f\left(\frac{1}{1-\delta}r\right)$ for any $r \geq 0$ and $\Delta = 2\varepsilon\sqrt{\frac{d}{d_\delta}} + 2\varepsilon(1 - \delta)$, then

937
$$|\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| \leq \omega(|\mathbf{y}_1 - \mathbf{y}_2| + \Delta), \quad \text{for any } \mathbf{y}_1, \mathbf{y}_2 \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}.$$

938 By Lemma 4.1, there exists \tilde{g} defined on \mathbb{R}^{d_δ} such that

939
$$|\tilde{g}(\mathbf{y}) - \tilde{f}(\mathbf{y})| \leq \omega(\Delta) = \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right), \quad \text{for any } \mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon), \quad (4.6)$$

940 and

941
$$|\tilde{g}(\mathbf{y}_1) - \tilde{g}(\mathbf{y}_2)| \leq \omega(|\mathbf{y}_1 - \mathbf{y}_2|) = \omega_f\left(\frac{1}{1-\delta}|\mathbf{y}_1 - \mathbf{y}_2|\right), \quad \text{for any } \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^{d_\delta}.$$

942 It follows that

943
$$\omega_{\tilde{g}}^{\mathbb{R}^{d_\delta}}(r) \leq \omega_f\left(\frac{r}{1-\delta}\right), \quad \text{for any } r \geq 0. \quad (4.7)$$

944 By Equation (4.4) and the definition of \mathcal{M}_ε in Equation (1.2), it is easy to check
 945 that

$$946 \quad \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbf{A}([0, 1]^d) \subseteq [-\sqrt{\frac{d}{d_\delta}}, \sqrt{\frac{d}{d_\delta}}]^{d_\delta}.$$

947 By the application of Theorem 4.3 with $E = [-\sqrt{\frac{d}{d_\delta}}, \sqrt{\frac{d}{d_\delta}}]^{d_\delta}$, there exists a function
 948 $\tilde{\phi}$ implemented by a ReLU FNN with width $3^{d_\delta+3} \max\{d_\delta \lfloor N^{1/d_\delta} \rfloor, N+1\}$ and depth
 949 $12L+14+2d_\delta$ such that

$$950 \quad \|\tilde{g} - \tilde{\phi}\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_{\tilde{g}}^E(2\sqrt{\frac{d}{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}). \quad (4.8)$$

951 Define $\phi := \tilde{\phi} \circ \mathbf{A}$, i.e., $\phi(\mathbf{x}) := \tilde{\phi}(\mathbf{Ax})$ for any $\mathbf{x} \in \mathbb{R}^d$. Then ϕ is also a ReLU FNN
 952 with width $3^{d_\delta+3} \max\{d_\delta \lfloor N^{1/d_\delta} \rfloor, N+1\}$ and depth $12L+14+2d_\delta$.

953 For any $\mathbf{x} \in \mathcal{M}_\varepsilon$, set $\mathbf{y} = \mathbf{Ax}$ and $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{z} \in \mathbb{R}^d : \mathbf{Az} = \mathbf{y}\})$, there exist $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_y \in \mathcal{M}$
 954 such that $|\tilde{\mathbf{x}} - \mathbf{x}| \leq \varepsilon$ and $|\tilde{\mathbf{x}}_y - \mathbf{x}_y| \leq \varepsilon$. It follows from Equation (4.5) that

$$\begin{aligned} |\mathbf{x} - \mathbf{x}_y| &\leq |\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_y| + 2\varepsilon \leq \frac{1}{1-\delta}|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}\tilde{\mathbf{x}}_y| + 2\varepsilon \\ 955 \quad &\leq \frac{1}{1-\delta}(|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{Ax}| + |\mathbf{Ax} - \mathbf{Ax}_y| + |\mathbf{Ax}_y - \mathbf{A}\tilde{\mathbf{x}}_y|) + 2\varepsilon \\ &= \frac{1}{1-\delta}(|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{Ax}| + |\mathbf{Ax}_y - \mathbf{A}\tilde{\mathbf{x}}_y|) + 2\varepsilon \leq \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon. \end{aligned} \quad (4.9)$$

956 In fact, the above equation implies that $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{Ax} = \mathbf{y}\}$ is contained in a small ball
 957 of radius $\mathcal{O}(\varepsilon)$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$ as we mentioned previously.

958 Together with Equation (4.6), (4.7), (4.8), and (4.9), we have, for any $\mathbf{x} \in \mathcal{M}_\varepsilon$,

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq |f(\mathbf{x}) - f(\mathbf{x}_y)| + |f(\mathbf{x}_y) - \phi(\mathbf{x})| \\ &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + |\tilde{f}(\mathbf{y}) - \tilde{\phi}(\mathbf{y})| \\ 959 \quad &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + |\tilde{f}(\mathbf{y}) - \tilde{g}(\mathbf{y})| + |\tilde{g}(\mathbf{y}) - \tilde{\phi}(\mathbf{y})| \\ &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_{\tilde{g}}^E(2\sqrt{\frac{d}{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}) \\ &\leq 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right). \end{aligned}$$

960 Hence, we have finished the proof of this theorem. \square

961 It is worth emphasizing that the approximation error

$$962 \quad \mathcal{O}\left(\omega_f(\mathcal{O}(\varepsilon)) + \omega_f(\mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta}))\right)$$

963 in Theorem 1.2 is equal to $\mathcal{O}\left(\omega_f(\mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta}))\right)$ when $\varepsilon = \mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta})$.

964 The application of Theorem 4.4 and the proof of Theorem 1.2 in fact inspire an
 965 efficient two-step algorithm for high-dimensional learning problems: in the first step,
 966 high-dimensional data are projected to a low-dimensional space via a random projection;
 967 in the second step, a deep learning algorithm is applied to learn from the low-dimensional
 968 data. By Theorem 4.4 and 1.2, the deep learning algorithm in the low-dimensional space
 969 can still provide good results with a high probability.

970 4.3 Optimal ReLU FNN structure in parallel computing

971 In this section, we show how to select the best ReLU FNN to approximate functions
 972 in $B_\lambda(C^\alpha([0, 1]^d))$ on a d -dimensional cube, if the approximation error ε and the number
 973 of parallel computing cores (processors) p are given. We choose the best ReLU FNN by
 974 minimizing the time complexity in each training iteration. The analysis in this section
 975 is valid up to a constant prefactor.

976 Assume $\phi_\theta \in \mathcal{NN}(\#\text{input} = d; \text{widthvec} = [N]^L; \#\text{output} = 1)$, $N, L \in \mathbb{N}^+$, where θ is
 977 the vector including all parameters of ϕ_θ . By the basic knowledge of parallel computing
 978 (see [36] for more details), we have the following Table 2.

Table 2: Time complexity of one training iteration for an FNN of width N and depth L .

Number of cores p	Time Complexity	
	Evaluating $\phi_\theta(\mathbf{x})$	Evaluating $\frac{\partial \phi_\theta(\mathbf{x})}{\partial \theta}$
$p \in [1, N]$	$\mathcal{O}(N^2 L/p)$	$\mathcal{O}(N^2 L/p)$
$p \in (N, N^2]$	$\mathcal{O}(L(N^2/p + \ln \frac{p}{N}))$	$\mathcal{O}(L(N^2/p + \ln \frac{p}{N}))$
$p \in (N^2, \infty)$	$\mathcal{O}(L \ln N)$	$\mathcal{O}(L \ln N)$

979 For the sake of simplicity, we assume that the training batch size is $\mathcal{O}(1)$. Denote
 980 the time complexity of each training iteration as $T(n, L)$, then

$$981 \quad T(N, L) = \begin{cases} \mathcal{O}(N^2 L/p), & p \in [1, N], \\ \mathcal{O}(L(N^2/p + \ln \frac{p}{N})), & p \in (N, N^2], \\ \mathcal{O}(L \ln N), & p \in (N^2, \infty). \end{cases}$$

982 Theorem 1.1 and 2.3 imply that the approximation error ε is essentially $\mathcal{O}((NL)^{-2\alpha/d})$.
 983 Hence, we can get the optimal size of ReLU FNNs via the optimization problem below:

$$984 \quad \begin{aligned} & (N_{\text{opt}}, L_{\text{opt}}) = \arg \min_{N, L} T(N, L) \\ & \text{subject to } \begin{cases} \varepsilon = \mathcal{O}((NL)^{-2\alpha/d}), \\ N, L, p \in \mathbb{N}^+. \end{cases} \end{aligned} \quad (4.10)$$

985 To simplify the discussion, we have the following assumptions:

- 986 • Dropping the notation $\mathcal{O}(\cdot)$ sometimes while assuming asymptotic analysis with
 987 the abuse of notations.
- 988 • N , L , and p are allowed to be real numbers.
- 989 • We denote $\varepsilon = (NL)^{-2\alpha/d}$ since the approximation rate $\mathcal{O}((NL)^{-2\alpha/d})$ is both at-
 990 tainable and nearly optimal.

991 With $\varepsilon = (NL)^{-2\alpha/d}$, we have

$$\begin{aligned}
\bar{T}(N, L) &:= \begin{cases} N^2L/p & p \in [1, N], \\ L(N^2/p + \ln \frac{p}{N}), & p \in (N, N^2], \\ L(1 + \ln N), & p \in [N^2, \infty), \end{cases} \\
&= \begin{cases} N\varepsilon^{-d/(2\alpha)}/p, & N \in [p, \infty), \\ N\varepsilon^{-d/(2\alpha)}/p + \frac{1}{N}\varepsilon^{-d/(2\alpha)} \ln \frac{p}{N}, & N \in [\sqrt{p}, p), \\ \frac{1+\ln N}{N}\varepsilon^{-d/(2\alpha)}, & N \in [1, \sqrt{p}). \end{cases}
\end{aligned} \tag{4.11}$$

993 Then we get $T(N, L) = \mathcal{O}(\bar{T}(N, L))$. Therefore, the optimization problem in Equation
994 (4.10) can be simplified to

$$\begin{aligned}
(N_{\text{opt}}, L_{\text{opt}}) &= \arg \min_{N, L} \bar{T}(N, L) \\
\text{subject to } &\begin{cases} \varepsilon = (NL)^{-2\alpha/d}, \\ N, L, p \in [1, \infty). \end{cases}
\end{aligned} \tag{4.12}$$

996 By Equation (4.11), $\bar{T}(N, L)$ is independent of L on the condition that $\varepsilon = (NL)^{-2\alpha/d}$.
997 Therefore, we may denote $\bar{T}(N, L)$ by $\bar{T}(N)$. Now we consider two cases: the case
998 $p = \mathcal{O}(1)$ and the case $p \gg \mathcal{O}(1)$.

999 **Case 1:** The case $p = \mathcal{O}(1)$.

1000 It is clear that $\bar{T}(N)$ is increasing in N when $N \in [p, \infty)$ by Equation (4.11).
1001 Together with $p = \mathcal{O}(1)$, then $\mathcal{O}(\sqrt{p}) = \mathcal{O}(p) = \mathcal{O}(1)$. Therefore, $N_{\text{opt}} = \mathcal{O}(1)$ and
1002 $L_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)})$. Note that we regard d as a constant ($\mathcal{O}(1)$) in above analysis, N_{opt}
1003 should be $\mathcal{O}(d)$ in fact.

1004 **Case 2:** The case $p \gg \mathcal{O}(1)$.

1005 Since $\varepsilon = (NL)^{-2\alpha/d}$, we have $N \leq \varepsilon^{-d/(2\alpha)}$. We only need to consider the monotonic-
1006 ity of $\bar{T}(N)$ on $[1, \varepsilon^{-d/(2\alpha)}]$. Together with Equation (4.11), this case can be divided into
1007 two sub-cases: the sub-case $\sqrt{p} \leq \varepsilon^{-d/(2\alpha)}$ and the sub-case $\sqrt{p} > \varepsilon^{-d/(2\alpha)}$.

1008 **Case 2.1:** The sub-case $\sqrt{p} > \varepsilon^{-d/(2\alpha)}$.

1009 $\sqrt{p} > \varepsilon^{-d/(2\alpha)}$ implies $[1, \varepsilon^{-d/(2\alpha)}] \subseteq [1, \sqrt{p}]$. Hence, $\bar{T}(N)$ is decreasing in N on
1010 $[1, \varepsilon^{-d/(2\alpha)}]$. It follows that $N_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)})$ and that $L_{\text{opt}} = \mathcal{O}(1)$.

1011 **Case 2.2:** The sub-case $\sqrt{p} \leq \varepsilon^{-d/(2\alpha)}$.

1012 For this sub-case, N_{opt} and N_{opt} are hard to estimate. However, we can give a
1013 rough range of N_{opt} . Since $\bar{T}(N)$ is decreasing in N on $[1, \sqrt{p}]$ and increasing in N on
1014 $[p, \infty)$, the minimum of $\bar{T}(N)$ is achieved on $[\sqrt{p}, p]$. Hence, $N_{\text{opt}} \in [\mathcal{O}(\sqrt{p}), \mathcal{O}(p)] \cap$
1015 $[\mathcal{O}(\sqrt{p}), \mathcal{O}(\varepsilon^{-d/(2\alpha)})]$ and $L_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)}/N_{\text{opt}})$.

1016 5 Conclusion and future work

1017 This paper aims at a quantitative and optimal approximation rate of ReLU FNNs
1018 in terms of both width and depth simultaneously to approximate continuous functions.

1019 It was shown that ReLU FNNs with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ can approximate
 1020 an arbitrary continuous function on a d -dimensional cube with an approximation rate
 1021 $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$. In particular, when f is a Hölder continuous function of order α
 1022 with a Hölder constant λ , the approximation rate is $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ and it is nearly
 1023 asymptotically tight. We also extended our analysis to the case when the domain of
 1024 f is irregular and showed the same approximation rate. In practical applications, it is
 1025 usually believed that real data are sampled from an ε -neighborhood of a $d_{\mathcal{M}}$ -dimensional
 1026 smooth manifold $\mathcal{M} \subseteq [0, 1]^d$ with $d_{\mathcal{M}} \ll d$. In the case of an essentially low-dimensional
 1027 domain, we show an approximation rate

$$1028 \quad 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right)$$

1029 for ReLU FNNs to approximate f in the ε -neighborhood, $d_\delta = \mathcal{O}\left(d_{\mathcal{M}}\frac{\ln(d/\delta)}{\delta^2}\right)$ for any given
 1030 $\delta \in (0, 1)$.

1031 Besides, we studied how to select the best ReLU FNN to approximate continuous
 1032 function in parallel computing. In particular, ReLU FNNs with depth $\mathcal{O}(1)$ are the best
 1033 choices if the number of parallel computing cores p is sufficiently large. ReLU FNNs
 1034 with width $\mathcal{O}(d)$ are best choices if $p = \mathcal{O}(1)$. The width of best ReLU FNNs is between
 1035 $\mathcal{O}(\sqrt{p})$ and $\mathcal{O}(p)$ if p is moderate.

1036 We would like to remark that our analysis was based on the fully connected feed-
 1037 forward neural networks and the ReLU activation function. It would be very interesting
 1038 to generalize our conclusions to neural networks with other types of architectures (e.g.,
 1039 convolutional neural networks) and activation functions (e.g., tanh and sigmoid func-
 1040 tions). Besides, if identity maps are allowed in the construction of neural networks as in
 1041 the residual networks [28], the size of FNNs in our construction can be further optimized.
 1042 Finally, the proposed analysis could be generalized to other function spaces with explicit
 1043 formulas to characterize the approximation error. These will be left as future work.

1044 Acknowledgments

1045 Z. Shen is supported by Tan Chin Tuan Centennial Professorship. H. Yang was
 1046 partially supported by the US National Science Foundation under award DMS-1945029.

1047 References

- 1048 [1] O. ABDEL-HAMID, A. MOHAMED, H. JIANG, L. DENG, G. PENN, AND D. YU,
 1049 *Convolutional neural networks for speech recognition*, IEEE/ACM Transactions on
 1050 Audio, Speech, and Language Processing, 22 (2014), pp. 1533–1545.
- 1051 [2] M. ANTHONY AND P. L. BARTLETT, *Neural Network Learning: Theoretical Foun-
 1052 dations*, Cambridge University Press, New York, NY, USA, 1st ed., 2009.
- 1053 [3] R. G. BARANIUK AND M. B. WAKIN, *Random projections of smooth manifolds*,
 1054 Foundations of Computational Mathematics, 9 (2009), pp. 51–77.

- 1055 [4] A. R. BARRON, *Universal approximation bounds for superpositions of a sigmoidal*
1056 *function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.
- 1057 [5] P. BARTLETT, V. MAIOROV, AND R. MEIR, *Almost linear VC dimension bounds*
1058 *for piecewise polynomial networks*, Neural Computation, 10 (1998), pp. 217–3.
- 1059 [6] M. BIANCHINI AND F. SCARSELLI, *On the complexity of neural network classifiers:*
1060 *A comparison between shallow and deep architectures*, IEEE Transactions on Neural
1061 Networks and Learning Systems, 25 (2014), pp. 1553–1565.
- 1062 [7] E. K. BLUM AND L. K. LI, *Approximation theory and feedforward networks*, Neural
1063 Networks, 4 (1991), pp. 511 – 515.
- 1064 [8] D. S. BROOMHEAD AND D. LOWE, *Multivariable Functional Interpolation and*
1065 *Adaptive Networks*, Complex Systems 2, (1988), pp. 321–355.
- 1066 [9] J. CAI, D. LI, J. SUN, AND K. WANG, *Enhanced expressive power and fast training*
1067 *of neural networks by random projections*, CoRR, abs/1811.09054 (2018).
- 1068 [10] S. CHEN AND D. DONOHO, *Basis pursuit*, in Proceedings of 1994 28th Asilomar
1069 Conference on Signals, Systems and Computers, vol. 1, Oct 1994, pp. 41–44 vol.1.
- 1070 [11] C. K. CHUI, S.-B. LIN, AND D.-X. ZHOU, *Construction of neural networks for re-*
1071 *alization of localized deep learning*, Frontiers in Applied Mathematics and Statistics,
1072 4 (2018), p. 14.
- 1073 [12] D. C. CIREŞAN, U. MEIER, J. MASCI, L. M. GAMBARDELLA, AND J. SCHMID-
1074 HUBER, *Flexible, high performance convolutional neural networks for image clas-*
1075 *sification*, in Proceedings of the Twenty-Second International Joint Conference on
1076 Artificial Intelligence - Volume Volume Two, IJCAI'11, AAAI Press, 2011, pp. 1237–
1077 1242.
- 1078 [13] D. COSTARELLI AND A. R. SAMBUCINI, *Saturation classes for max-product neu-*
1079 *ral network operators activated by sigmoidal functions*, Results in Mathematics, 72
1080 (2017), pp. 1555 – 1569.
- 1081 [14] D. COSTARELLI AND G. VINTI, *Convergence for a family of neural network oper-*
1082 *ators in orlicz spaces*, Mathematische Nachrichten, 290 (2017), pp. 226–235.
- 1083 [15] —, *Approximation results in orlicz spaces for sequences of kantorovich max-*
1084 *product neural network operators*, Results in Mathematics, 73 (2018), pp. 1 – 15.
- 1085 [16] G. CYBENKO, *Approximation by superpositions of a sigmoidal function*, MCSS, 2
1086 (1989), pp. 303–314.
- 1087 [17] I. DAUBECHIES, R. DEVORE, S. FOUCART, B. HANIN, AND G. PETROVA, *Non-*
1088 *linear approximation and (deep) ReLU networks*, vol. abs/1905.02199, 2019.

- 1089 [18] R. DEVORE AND A. RON, *Approximation using scattered shifts of a multivariate*
1090 *function*, Transactions of the American Mathematical Society, 362 (2010), pp. 6205–
1091 6229.
- 1092 [19] R. A. DEVORE, *Nonlinear approximation*, Acta Numerica, 7 (1998), p. 51–150.
- 1093 [20] W. E, J. HAN, AND A. JENTZEN, *Deep learning-based numerical methods for high-*
1094 *dimensional parabolic partial differential equations and backward stochastic differ-*
1095 *ential equations*, Communications in Mathematics and Statistics, 5 (2017), pp. 349–
1096 380.
- 1097 [21] W. E, C. MA, AND Q. WANG, *A priori estimates of the population risk for residual*
1098 *networks*, ArXiv, abs/1903.02154 (2019).
- 1099 [22] W. E, C. MA, AND L. WU, *A priori estimates of the population risk for two-layer*
1100 *neural networks*, Communications in Mathematical Sciences, 17 (2019), pp. 1407 –
1101 1425.
- 1102 [23] W. E AND Q. WANG, *Exponential convergence of the deep neural network approx-*
1103 *imation for analytic functions*, CoRR, abs/1807.00297 (2018).
- 1104 [24] J. HAN, A. JENTZEN, AND W. E, *Solving high-dimensional partial differential*
1105 *equations using deep learning*, Proceedings of the National Academy of Sciences,
1106 115 (2018), pp. 8505–8510.
- 1107 [25] T. HANGELBROEK AND A. RON, *Nonlinear approximation using gaussian kernels*,
1108 Journal of Functional Analysis, 259 (2010), pp. 203 – 219.
- 1109 [26] B. HANIN AND M. SELLKE, *Approximating continuous functions by ReLU nets of*
1110 *minimal width*, (2017).
- 1111 [27] N. HARVEY, C. LIAW, AND A. MEHRABIAN, *Nearly-tight VC-dimension bounds*
1112 *for piecewise linear neural networks*, in Proceedings of the 2017 Conference on Learn-
1113 ing Theory, S. Kale and O. Shamir, eds., vol. 65 of Proceedings of Machine Learning
1114 Research, Amsterdam, Netherlands, 07–10 Jul 2017, PMLR, pp. 1064–1068.
- 1115 [28] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recog-*
1116 *niton*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition
1117 (CVPR), June 2016, pp. 770–778.
- 1118 [29] K. HORNIK, *Approximation capabilities of multilayer feedforward networks*, Neural
1119 Networks, 4 (1991), pp. 251 – 257.
- 1120 [30] K. HORNIK, M. STINCHCOMBE, AND H. WHITE, *Multilayer feedforward networks*
1121 *are universal approximators*, Neural Networks, 2 (1989), pp. 359 – 366.

- 1122 [31] M. HUTZENTHALER, A. JENTZEN, T. KRUSE, AND T. A. NGUYEN, *A proof that*
1123 *rectified deep neural networks overcome the curse of dimensionality in the numerical*
1124 *approximation of semilinear heat equations*, SN Partial Differential Equations and
1125 Applications, (2020).
- 1126 [32] K. KAWAGUCHI, *Deep learning without poor local minima*, in Advances in Neu-
1127 ral Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg,
1128 I. Guyon, and R. Garnett, eds., Curran Associates, Inc., 2016, pp. 586–594.
- 1129 [33] K. KAWAGUCHI AND Y. BENGIO, *Depth with nonlinearity creates no bad local*
1130 *minima in resnets*, (2018).
- 1131 [34] M. J. KEARNS AND R. E. SCHAPIRE, *Efficient distribution-free learning of prob-*
1132 *abilistic concepts*, J. Comput. Syst. Sci., 48 (1994), pp. 464–497.
- 1133 [35] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with*
1134 *deep convolutional neural networks*, in Advances in Neural Information Processing
1135 Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.,
1136 Curran Associates, Inc., 2012, pp. 1097–1105.
- 1137 [36] V. KUMAR, *Introduction to Parallel Computing*, Addison-Wesley Longman Pub-
1138 lishing Co., Inc., Boston, MA, USA, 2nd ed., 2002.
- 1139 [37] V. KŮRKOVÁ, *Kolmogorov’s theorem and multilayer neural networks*, Neural Net-
1140 works, 5 (1992), pp. 501 – 506.
- 1141 [38] G. LEWICKI AND G. MARINO, *Approximation of functions of finite variation by*
1142 *superpositions of a sigmoidal function*, Applied Mathematics Letters, 17 (2004),
1143 pp. 1147 – 1152.
- 1144 [39] S. LIANG AND R. SRIKANT, *Why deep neural networks?*, CoRR, abs/1610.04161
1145 (2016).
- 1146 [40] S. LIN, X. LIU, Y. RONG, AND Z. XU, *Almost optimal estimates for approxima-*
1147 *tion and learning by radial basis function networks*, Machine Learning, 95 (2014),
1148 pp. 147–164.
- 1149 [41] B. LLANAS AND F. SAINZ, *Constructive approximate interpolation by neural net-*
1150 *works*, Journal of Computational and Applied Mathematics, 188 (2006), pp. 283 –
1151 308.
- 1152 [42] J. LU, Z. SHEN, H. YANG, AND S. ZHANG, *Deep Network Approximation for*
1153 *Smooth Functions*, arXiv e-prints, (2020), p. arXiv:2001.03040.
- 1154 [43] Z. LU, H. PU, F. WANG, Z. HU, AND L. WANG, *The expressive power of neural*
1155 *networks: A view from the width*, in Advances in Neural Information Processing
1156 Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-
1157 wanathan, and R. Garnett, eds., Curran Associates, Inc., 2017, pp. 6231–6239.

- 1158 [44] V. MAIOROV AND A. PINKUS, *Lower bounds for approximation by mlp neural*
1159 *networks*, Neurocomputing, 25 (1999), pp. 81 – 91.
- 1160 [45] S. G. MALLAT AND Z. ZHANG, *Matching pursuits with time-frequency dictionaries*,
1161 IEEE Transactions on Signal Processing, 41 (1993), pp. 3397–3415.
- 1162 [46] E. J. MCSHANE, *Extension of range of functions*, Bull. Amer. Math. Soc., 40
1163 (1934), pp. 837–842.
- 1164 [47] H. MONTANELLI AND Q. DU, *New error bounds for deep ReLU networks using*
1165 *sparse grids*, SIAM Journal on Mathematics of Data Science, 1 (2019), pp. 78–92.
- 1166 [48] H. MONTANELLI AND H. YANG, *Error bounds for deep ReLU networks using the*
1167 *kolmogorov–arnold superposition theorem*, Neural Networks, 129 (2020), pp. 1 – 6.
- 1168 [49] H. MONTANELLI, H. YANG, AND Q. DU, *Deep ReLU networks overcome the curse*
1169 *of dimensionality for bandlimited functions*, Journal of Computational Mathematics,
1170 (to appear).
- 1171 [50] G. F. MONTUFAR, R. PASCANU, K. CHO, AND Y. BENGIO, *On the number of*
1172 *linear regions of deep neural networks*, in Advances in Neural Information Processing
1173 Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q.
1174 Weinberger, eds., Curran Associates, Inc., 2014, pp. 2924–2932.
- 1175 [51] Q. N. NGUYEN AND M. HEIN, *The loss surface of deep and wide neural networks*,
1176 CoRR, abs/1704.08045 (2017).
- 1177 [52] J. PARK AND I. W. SANDBERG, *Universal approximation using radial-basis-*
1178 *function networks*, Neural Computation, 3 (1991), pp. 246–257.
- 1179 [53] P. PETERSEN AND F. VOIGTLAENDER, *Optimal approximation of piecewise smooth*
1180 *functions using deep ReLU neural networks*, Neural Networks, 108 (2018), pp. 296
1181 – 330.
- 1182 [54] P. PETRUSHEV, *Multivariate n -term rational and piecewise polynomial approxima-*
1183 *tion*, Journal of Approximation Theory, 121 (2003), pp. 158 – 197.
- 1184 [55] D. ROLNICK AND M. TEGMARK, *The power of deeper networks for expressing*
1185 *natural functions*, CoRR, abs/1705.05502 (2017).
- 1186 [56] I. SAFRAN AND O. SHAMIR, *Depth-width tradeoffs in approximating natural func-*
1187 *tions with neural networks*, in Proceedings of the 34th International Conference on
1188 Machine Learning, D. Precup and Y. W. Teh, eds., vol. 70 of Proceedings of Machine
1189 Learning Research, International Convention Centre, Sydney, Australia, 06–11 Aug
1190 2017, PMLR, pp. 2979–2987.
- 1191 [57] A. SAKURAI, *Tight bounds for the VC-dimension of piecewise polynomial networks*,
1192 in Advances in Neural Information Processing Systems, Neural information process-
1193 ing systems foundation, 1999, pp. 323–329.

- 1194 [58] D. SCHERER, A. MÜLLER, AND S. BEHNKE, *Evaluation of pooling operations in*
1195 *convolutional architectures for object recognition*, in Artificial Neural Networks –
1196 ICANN 2010, K. Diamantaras, W. Duch, and L. S. Iliadis, eds., Berlin, Heidelberg,
1197 2010, Springer Berlin Heidelberg, pp. 92–101.
- 1198 [59] J. SCHMIDT-HIEBER, *Nonparametric regression using deep neural networks with*
1199 *ReLU activation function*, Ann. Statist., 48 (2020), pp. 1875–1897.
- 1200 [60] U. SHAHAM, A. CLONINGER, AND R. R. COIFMAN, *Provable approximation prop-*
1201 *erties for deep neural networks*, Applied and Computational Harmonic Analysis, 44
1202 (2018), pp. 537 – 557.
- 1203 [61] Z. SHEN, H. YANG, AND S. ZHANG, *Nonlinear approximation via compositions*,
1204 Neural Networks, 119 (2019), pp. 74 – 84.
- 1205 [62] T. SUZUKI, *Adaptivity of deep ReLU network for learning in besov and mixed smooth*
1206 *besov spaces: optimal rate and curse of dimensionality*, in International Conference
1207 on Learning Representations, 2019.
- 1208 [63] H. WHITNEY, *Analytic extensions of differentiable functions defined in closed sets*,
1209 Transactions of the American Mathematical Society, 36 (1934), pp. 63–89.
- 1210 [64] T. F. XIE AND F. L. CAO, *The rate of approximation of gaussian radial basis*
1211 *neural networks in continuous function space*, Acta Mathematica Sinica, English
1212 Series, 29 (2013), pp. 295–302.
- 1213 [65] D. YAROTSKY, *Error bounds for approximations with deep ReLU networks*, Neural
1214 Networks, 94 (2017), pp. 103 – 114.
- 1215 [66] D. YAROTSKY, *Optimal approximation of continuous functions by very deep ReLU*
1216 *networks*, in Proceedings of the 31st Conference On Learning Theory, S. Bubeck,
1217 V. Perchet, and P. Rigollet, eds., vol. 75 of Proceedings of Machine Learning Re-
1218 search, PMLR, 06–09 Jul 2018, pp. 639–649.