ReLU Network with Width $d + \mathcal{O}(1)$ Can Achieve Optimal Approximation Rate

Chenghao Liu [1] **Minghua Chen** [1]

Abstract

The prevalent employment of narrow neural networks, characterized by their minimal parameter count per layer, has led to a surge in research exploring their potential as universal function approximators. A notable result in this field states that networks with just a width of $d + 1$ can approximate any continuous function for input dimension d arbitrarily well. However, the optimal approximation rate for these narrowest networks, i.e., the optimal relation between the count of tunable parameters and the approximation error, remained unclear. In this paper, we address this gap by proving that ReLU networks with width $d + 1$ can achieve the optimal approximation rate for continuous functions over the domain $[0, 1]^d$ under L^p norm for $p \in [1, \infty)$. We further show that for the uniform norm, a width of $d + 11$ is sufficient. We also extend the results to narrow feed-forward networks with various activations, confirming their capability to approximate at the optimal rate. This work adds to the understanding of universal approximation of narrow networks.

1. Introduction and Main Results

Neural networks have emerged as a key component in deep learning, garnering considerable interest due to their remarkable success in practice. Meanwhile, understanding the expressive power of neural networks is important for deep learning, and boasts a rich and extensive history.

The study of the universal approximation property can at least date back to the 1980s. The classical universal theorem holds for wide and shallow networks. Specifically, in the early years, researchers (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991; Leshno et al., 1993; Pinkus, 1999) show that networks with reasonable activation with two hidden layers can approximate any multivariate continuous function and Lebesgue-integrable function over a compact domain to any desired accuracy as the width goes to infinity.

Since 2010, deep learning has experienced accelerated development, sparking increased interest in understanding the expressive capabilities of neural networks. Although neural networks are recognized for their universal approximation capability, an essential question remains: what is the minimum number of parameters required for a neural network to approximate a function adequately? To address this query, researchers have primarily focused on establishing the upper and lower bounds for the size of both deep and shallow networks in the approximation of certain functions (Eldan & Shamir, 2016; Liang & Srikant, 2016; Telgarsky, 2016; Yarotsky, 2017), consistently highlighting the benefits of deeper architectures. Recently, (Yarotsky, 2018; Shen et al., 2019a; Yarotsky & Zhevnerchuk, 2020; Shen et al., 2022b) quantitatively characterizes the approximation capabilities of ReLU FNNs with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ by examining the approximation rate, that is, they show how fast the error $\inf_{g_{N,L}} \|f - g_{N,L}\|$ will decay in terms of N and L when approximating a given continuous or smooth function f and finally they find the optimal rate.

Meanwhile, the use of narrow network structures has become increasingly prevalent in practice. Typically, the width of a neural network is chosen to be near the input dimension, a preference driven by the reduced number of tunable parameters in training. However, the upper bounds or approximation rate results discussed above can not apply to these very narrow networks. For instance, it has been shown that for a ReLU network constrained to a width not exceeding the input dimension d, there exists a continuous function f from which the network cannot reduce its approximation error, regardless of the number of parameters (Hanin & Sellke, 2017).

Thus, some researchers shifted their focus to understanding the approximation capabilities of networks from the view of width (Lu et al., 2017; Hanin & Sellke, 2017; Kidger & Lyons, 2020; Park et al., 2020; Cai, 2022; Kim et al., 2023; Duan et al., 2023). This line of work is concerned with the minimal width required for a network to universally approximate a continuous mapping from a compact domain in $\mathbb{R}^d$ to $\mathbb{R}^v$. The goal of this research has been the determination of the exact minimal width $w_{\min}$ neces-

[1]School of Data Science, City University of Hong Kong. Correspondence to: Minghua Chen <minghua.chen@cityu.edu.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Table 1. A summary of known universal approximation results of FNNs with fixed width and depth $\mathcal{O}(L)$. For a target function class $\mathcal{F}$ and a hypothesis model (the set of networks) $\mathcal{H}_L$ with L measuring its complexity, the approximation rate quantifies the speed of convergence of the error $\sup_{f\in\mathcal{F}} \inf_{h\in\mathcal{H}_L} \|f-h\|$ in terms of L where the norm is associated with the target function class.

Reference	Function Class	Activation	Width	Approximation Rate	Optimality
(Hanin & Sellke, 2017)	$C([0,1]^d;\mathbb{R}^v)$	ReLU	$d+v$	$\mathcal{O}\left(\omega_f\left(L^{-1/d}\right)\right)$	Suboptimal
(Yarotsky, 2018)	$C([0,1]^d;\mathbb{R})$	ReLU	$2d+10$	$\mathcal{O}\left(\omega_f\left(L^{-2/d}\right)\right)$	Optimal
(Kidger & Lyons, 2020)	$C([0,1]^d;\mathbb{R}^v)$ $L^p(\mathbb{R}^d;\mathbb{R}^v)$	Continuous† ReLU	$d+v+1$ $d+v+1$	N.A N.A	
(Park et al., 2020)	$C([0,1]^d;\mathbb{R}^v)$ $L^p(\mathbb{R}^d;\mathbb{R}^v)$	ReLU+STEP ReLU	$\max\{d+1,v\}$ $\max\{d+1,v\}$	$\mathcal{O}\left(\omega_f\left(L^{-1/d}\right)\right)$	Suboptimal
(Cai, 2022)	$L^p([0,1]^d;\mathbb{R}^v)$ $C([0,1]^d;\mathbb{R}^v)$	LeakyReLU ReLU+Floor	$\max\{d,v,2\}$ $\max\{d,v,2\}$	N.A N.A	
(Duan et al., 2023)	$C([0,1]^d;\mathbb{R}^v)$	Leaky-ReLU	$\max\{d+1,v\}$	N.A	
This paper (Thm. 1.1)	$L^p([0,1]^d;\mathbb{R})$	ReLU	$\max\{d+1,5\}$	$\mathcal{O}\left(\omega_f\left(L^{-2/d}\right)\right)$	Optimal
This paper (Thm. 3.1)	$L^p([0,1]^d;\mathbb{R}^v)$	ReLU	$\max\{d+1,v+6\}$		
This paper (Thm. 3.1)	$C([0,1]^d;\mathbb{R}^v)$	ReLU	$d+v+10$	$\mathcal{O}\left(\omega_f\left(L^{-2/(d+1)}\right)\right)$	Nearly Optimal

† The activation function is assumed to be nonpolynomial and continuously differentiable at at least one point, with a nonzero derivative at that point.

sary for an FNN with some reasonable activation function to achieve universality. This width is established to be $\max\{d,v\}$ for L^p-integrable mappings from $[0,1]^d$ to $\mathbb{R}^v$, where $p\in[1,\infty)$, and $\max\{d+1,v\}$ for continuous function spaces $C([0,1]^d;\mathbb{R}^v)$. However, the approximation rate of these networks, with their minimal widths, has been either neglected due to the complexity of the methods involved or not optimally characterized. Our contribution addresses this oversight, providing a detailed analysis of the approximation rate for minimally wide networks (Theorems 1.1 and 3.1). Refer to Table 1 for a succinct overview of these developments.

Motivation: why we matter the approximation rate. The approximation rate is a fundamental metric that quantifies the efficacy of neural networks in representing various functions. An optimal rate within the specific class of ReLU networks denotes that a given neural network architecture is capable of harnessing the fullest potential for function approximation. The practical implications of this are profound. By demonstrating that a simple, yet commonly employed neural network structure, such as the narrow networks we examine, can realize the optimal approximation rate, we establish that these accessible and computationally efficient architectures do not compromise their ability to approximate functions. This, in turn, suggests that practitioners can confidently use these simpler networks without fearing a trade-off in performance, thus bridging the gap between theoretical optimality and practical utility.

1.1. Main Results and Contributions

We denote by $C\left([0,1]^d\right)$ the set of continuous functions over $[0,1]^d$ under uniform norm $\|f\|_{L^\infty([0,1]^d)} = \max_{x\in[0,1]^d}|f(x)|$ and by $L^p\left([0,1]^d\right)$ the set of L^p-integrable functions over $[0,1]^d$ under norm $\|f\|_{L^p([0,1]^d)} = \left(\int_{[0,1]^d}|f(x)|^p dx\right)^{1/p} < \infty$. *Here, without any specific implication, we always assume* $1 \le p < \infty$. We define the modulus of continuity of a continuous function $f\in C\left([0,1]^d\right)$ via

$$\omega_f(t) := \sup\left\{|f(\mathbf{x})-f(\mathbf{y})| : \|\mathbf{x}-\mathbf{y}\|_2 \le t, \mathbf{x},\mathbf{y}\in[0,1]^d\right\}$$

for any $t\ge 0$. Note that ω_f is well defined for any continuous function f but may not for Lebesgue-integrable functions. *Thus, when we use the modulus of continuity to characterize the rate for approximating Lebesgue-integrable functions, we may consider approximating continuous functions under L^p norm since the continuous function class is dense in the Lebesgue-integrable function class over any compact domain under L^p norm (Walter, 1987).* Our main result is that ReLU FNN with the minimum width to satisfy the universality can achieve the optimal rate, as shown below where the proof of Theorem 1.1 is deferred to Appendix B for (i) and Appendix C for (ii).

Theorem 1.1. *Let $d\in\mathbb{N}$. For any continuous function $f\in C([0,1]^d)$, we have the following approximation results: (i) for $p\in[1,\infty)$, there exists a ReLU neural network $\rho:\mathbb{R}^d\to\mathbb{R}$ with width $\max\{d+1,5\}$ and depth not more*

than $25L + 7d + 4$ *such that*

$$\|f - \rho\|_{L^p([0,1]^d)} \leq 7\sqrt{d}\omega_f(L^{-\frac{2}{d}});$$

(ii) for $p = \infty$*, there exists a ReLU neural network* $\rho : \mathbb{R}^d \to \mathbb{R}$ *with width* $d + 11$ *and depth not more than* $6d3^dL$ *such that*

$$\|f - \rho\|_{L^\infty([0,1]^d)} \leq 3^d(3d + 1)(6d^{\frac{3}{2}} + 1)\omega_f(L^{-\frac{2}{d+1}}).$$

Our main contributions are summarized as follows:

▷ While the minimum width of ReLU FNNs to achieve the universality is known, we extend the approximate rate of ReLU FNN with the minimum width to optimal (Theorem 1.1). Besides, we have a more general result in Theorem 3.1 which extends the target functions to mappings from $[0, 1]^d$ to $\mathbb{R}$. Our contribution completes the approximation rate map of ReLU FNNs as illustrated in Figure 1, ensuring that the approximation rates for continuous functions across various regions, denoted by distinct colors, can not be imporved.

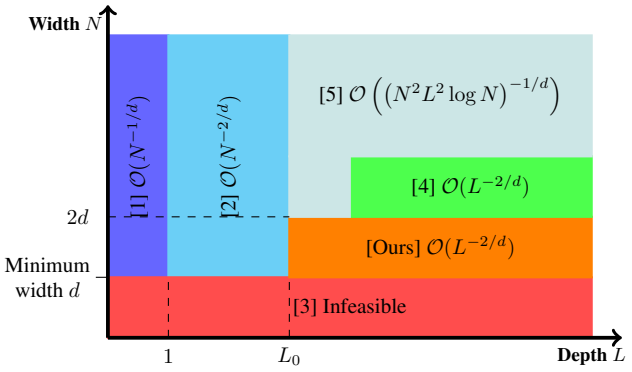

Figure 1. A summary of existing and our new results on the approximation rate of narrow ReLU FNNs with width $d + \mathcal{O}(1)$ and depth L for continuous functions. The existing results can be found from [1][2][4] (Yarotsky, 2018; Shen et al., 2019b), [5] (Shen et al., 2022b), and [3] (Hanin & Sellke, 2017; Park et al., 2020; Cai, 2022).

▷ Building on the results in Theorem 1.1, we show that narrow FNNs with either variant of ReLU functions (e.g. $\mathrm{LeakyReLU}, \mathrm{ELU}, \mathrm{ReLU}^2$) or other commonly used activation functions (e.g. $\mathrm{Sigmoid}, \mathrm{Tanh}, \mathrm{Softsign}$) can also achieve this enhanced approximation rate. Specifically, networks employing some ReLU variants such as $\mathrm{ELU}, \mathrm{Softplus}, \mathrm{Mish}$, network with nearly minimum width to satisfy the universality, i.e., $d + 1$ (or $d + \mathcal{O}(1)$), can achieve the approximation rate in Theorem 1.1. Further, we confirm that the rate of $\mathcal{O}(L^{-2/d})$ in Theorem 1.1 on the approximation of continuous functions is also optimal for narrow networks with activation functions like $\mathrm{LeakyReLU}, \mathrm{ReLU}^2, \mathrm{Softsign}$.

2. Related Work

In recent years, the expressive power of diverse neural network architectures has attracted wide interest, propelled by their impressive and noteworthy achievements in numerous domains. In this section, we focus on the function approximation perspective, offering an overview of previous work relevant to our study.

2.1. Universal Approximation Property

Universal approximation property of a function family $\mathcal{H}$ implies that $\mathcal{H}$ is dense in the continuous space $C([0,1]^d)$/$L^p([0,1]^d)$, i.e., for $f \in C([0,1]^d)$/$L^p([0,1]^d)$ and any desired error ϵ, there is $h \in \mathcal{H}$ such that $\|h - f\| \leq \epsilon$. The universal approximation property has been widely studied from shallow and wide networks with suitable activation functions (e.g., sigmoid, non-polynomial) (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991; Leshno et al., 1993) to its dual scenario, narrow and deep networks (e.g., (Hanin & Sellke, 2017; Kidger & Lyons, 2020)). Over the past ten years, various network architectures have been developed to cater to diverse tasks and objectives, aside from FNNs. The universal approximation property has been studied for various network architectures, such as 1D convolutional neural networks (CNNs) (Zhou, 2018; 2020b), 2D CNNs with the classical structure (He et al., 2022), continuous-time recurrent neural network (Li et al., 2020; 2022b), continuous-time ResNet (Li et al., 2022a), and ResNet with one neuron per hidden layer (Lin & Jegelka, 2018).

Approaches. Classical results, e.g., (Cybenko, 1989; Hornik et al., 1989), utilize real analysis but not constructive methods to demonstrate their result, hence it is not available for the actual approximation rate from these results. It is well known that polynomials, as per Stone-Weierstrass theorem, or piecewise linear functions are dense in continuous function space and step functions are dense in Lebesgue-integrable function space. Thus, to show the universality of some NN architecture, modern approaches often construct the architecture to generate or approximate a dense class of $C([0,1]^d)$/$L^p([0,1]^d)$. This technical extends beyond real-valued continuous function approximators to complex-valued function approximators e.g.,(Geuchen et al., 2023; Voigtlaender, 2023), and to permutation invariant function approximators e.g.,(Segol & Lipman, 2019; Sannai et al., 2019), among others. Furthermore, the quantitative version of Stone-Weierstrass theorem and spline approximation theory inform us that any continuous function f over $[0,1]^d$ can be approximated by a polynomial p_n with degree $\mathcal{O}(n)$ and a piecewise linear interpolation function g_n on the grid $(\mathbb{Z}/n)^d$ such that $\|f - p_n\|_\infty = \mathcal{O}(\omega_f(1/\sqrt{n}))$ (Kratsios & Papon, 2022) and $\|f - g_n\|_\infty = \mathcal{O}(\omega_f(1/n))$ (Yarotsky, 2018). Thus, if the size of a model class to approximate a polynomial or piecewise linear function is known, one

can deduce the approximation rate of that model class for continuous functions.

2.2. Minimum Width of FNNs to Satisfy Universality

While classical approximation theory most focuses on 'fat' (wide and shallow) networks, recent research become more interested in its dual scenario–'flat' (narrow and deep) networks and their universality has been explored extensively. We denote $L^p([0,1]^d;\mathbb{R}^v)/C([0,1]^d;\mathbb{R}^v)$ by the set of mappings from $[0,1]^d$ to $\mathbb{R}^v$ and use $w_{\min}$ to denote the minimum width of FNNs to possess the universal approximation property. For function approximation in $L^p(\mathbb{R}^d;\mathbb{R})$, the lower bound of $w_{\min}$ is $d+1$ for ReLU FNNs given by (Lu et al., 2017). When focusing on a compact domain, specifically for approximating functions in $L^p([0,1]^d;\mathbb{R})$, the lower bound decreases to d (Lu et al., 2017). When extending to approximate mappings in $L^p(\mathbb{R}^d;\mathbb{R})$ and $L^p([0,1]^d;\mathbb{R}^v)$, the lower bound of $w_{\min}$ is $\max\{d+1,v\}$ and $\max\{d,v\}$ respectively for ReLU networks (Park et al., 2020), and also hold for networks with arbitrary activation (Cai, 2022). Recent work (Park et al., 2020) and (Kim et al., 2023) has pinpointed the tight upper bounds of $w_{\min}$ for ReLU(-Like) networks approximating mappings in $L^p([0,1]^d;\mathbb{R}^v)$ and $L^p(\mathbb{R}^d;\mathbb{R}^v)$ to be $\max\{d+1,v\}$ and $\max\{d,v,2\}$ respectively, thus, precisely determining the minimum width on the approximation in L^p space. Later, for LeakyReLU networks, (Cai, 2022) find the exact $w_{\min}$ in L^p space. The approximation of continuous mappings in $C([0,1]^d;\mathbb{R}^v)$ presents additional complexities. Its lower bound of $w_{\min}$ is eastablished at both $d+1$ (Hanin & Sellke, 2017) and $v+\mathbb{1}_{\{d<v\leq 2d\}}$ (Kim et al., 2023) for ReLU FNNs, and at $\max d, v$ for FNN with arbitrary activation (Cai, 2022). The upper bounds of $w_{\min}$ on approximation of space $C([0,1]^d;\mathbb{R}^v)$ is $d+v$ for ReLU FNNs (Hanin & Sellke, 2017) and $d+v+1$ for FNNs with continuous activation under some mild condition (Kidger & Lyons, 2020). While (Cai, 2022) shows that the upper bound of $w_{\min}$ for Leaky-ReLU networks can be lowered to $\max\{d,v,2\}$ which is optimal, it remains an open question that if the upper bound can be lowered for ReLU networks. Nonetheless, on the approximation of continuous functions over $[0,1]^d$, the minimum width to satisfy the universality is exactly $d+1$. For those interested, a comprehensive summary of these results can be found in Table 1 presented in (Kim et al., 2023).

Approaches. On approximating mappings from a compact domain over $\mathbb{R}^d$ to $\mathbb{R}^v$, initial approaches (Lu et al., 2017; Hanin & Sellke, 2017; Kidger & Lyons, 2020) typically allocate d neurons in each layer to forward the value of the input and additional $v+c$ neurons (with c being a small constant) for intermediate computations. They construct this kind of narrow network to approximate piecewise linear/constant functions or polynomials to show its universality. As we will further discuss later in Subsection 2.3 their constructive network parameter, regarded as a mapping over the target function space, is a continuous mapping. Advancements were marked by (Park et al., 2020; Kim et al., 2023), who introduce an encoding-memorizing-decoding scheme to effectively lower down the upper bound from $d+v+c$ to $\max\{d,v(+1)\}$. Specifically, this method begins by partitioning the interval into numerous segments and mapping the input x to the leftmost endpoint of the corresponding segment, thus transforming the input into a finite set. They then decode the finite set to a scalar number. Following this, they employ v neurons in each layer to compute the encoder number of $f(x)$. Finally, they decode the number back into the v output values. Their constructive method is in a discontinuous phase and the approximation rate is not optimal. Recently, (Cai, 2022) found the exact minimum width $w_{\min}$ for LeakyReLU networks by showing that the narrow LeakyReLU network can approximate any flow map that can approximate the target mapping. However, the approximation rate is not available due to the implicit rate of flow map approximation.

2.3. Approximation Rate of FNNs

The approximation rate precisely measures the effectiveness of an approximation. In other words, given a target function f and a hypothesis model $\mathcal{H}_n$ where n denote the complexity of the model, it characterized how fast the error $\inf_{g_n\in\mathcal{H}_n}\|f-g_n\|$ will decay as n increases. The approximation rate of FNNs which is a special case of nonlinear approximation, has been widely studied and has a long historical origin.

Optimality. To ascertain whether a model class attains an optimal approximation rate, it is imperative to identify the lower bound of the approximation rate. The lower bound of the approximation rate of continuous approximators is limited by metric entropy (Kolmogorov & Tikhomirov, 1959). For a compact set $\mathcal{C}$ over a metric space $\mathcal{M}$, the ϵ metric entropy of $\mathcal{C}$ is a number $\log N(\epsilon;\mathcal{C})$ where $N(\epsilon;\mathcal{C})$ is the cardinality of the smallest ϵ-covering, i.e.,

$$\mathcal{C}\subseteq\bigcup_{i=1}^{N(\epsilon;\mathcal{C})}B\left(x_i,\epsilon\right)$$

and $B(x_i,\epsilon)$ is the ball in the metric space $\mathcal{M}$ centered at x_i with radius ϵ. Thus, metric entropy reflects the smallest number of bits needed to approximate any element $x\in\mathcal{C}$. It has been shown that the metric entropy of the Lipschitz continuous function space $\mathrm{Lip}([0,1]^d;\mu)$ under uniform metric is $\Theta((\mu/\epsilon)^d)$ (DeVore et al., 1989). Given an approximator $\mathcal{H}$ with a complexity measure n, and a mapping τ from $\mathbb{R}^n\to\mathcal{H}$, if the approximation rate of $\mathcal{H}$ for Lipschitz

continuous functions satisfy

$$\sup_{f\in \mathrm{Lip}([0,1]^d)} \inf_{F:\mathrm{Lip}([0,1]^d)\to\mathbb{R}^n} \|f-\tau(F(f))\|_\infty = \mathcal{O}(n^{-k}) \tag{1}$$

for some positive number k, then we have that $k \leq 1/d$ under the assumption that F is continuous meaning that $\mathcal{H}$ is an continuous approximator, according to (DeVore et al., 1989; Yarotsky, 2017). Notably, setting $\mathcal{O}(n^{-1/d}) \leq \epsilon$, we have $n \geq \Theta(\epsilon^{-d})$. Thus, the approximation rate (1) of a continuous approximator, not only FNN, is inherently limited by the metric entropy. However, a discontinuous approximator could yield a distinct result. If we regard FNN as a discontinuous approximator, i.e., $F : \mathrm{Lip}([0,1]^d) \to \mathbb{R}^n$ in (1) is discontinuous where $\mathbb{R}^n$ is the parameter space and τ is the structure of FNN, then the approximation rate is limited by Vapnik-Chervonenkis (VC) dimension (Goldberg & Jerrum, 1993). Consequently, the approximation rate (1) of ReLU FNN with n parameters is lower bounded by $k \leq 2/d$ (Shen et al., 2022b; Yarotsky, 2018; Yarotsky & Zhevnerchuk, 2020) which is the result of VC dimension of ReLU FNN (Bartlett et al., 2019; Anthony et al., 1999).

Therefore, researchers hope that FNN can achieve this optimal approximation rate in Lipschitz function space. Pioneering work (Yarotsky, 2017) constructively shows that ReLU deep FNNs can become optimal continuous approximators for Lipschitz functions. Following this work, (Yarotsky, 2018; Yarotsky & Zhevnerchuk, 2020) extend the optimality to ReLU FNN discontinuous approximator and (Shen et al., 2019a; Lu et al., 2021; Shen et al., 2022b) later achieve the optimal approximation by ReLU FNN in terms of width and depth.

Meanwhile, there is an ongoing effort to attain a higher approximation rate using FNN which will break the limit of the metric entropy. In the beginning, they search for smaller function space, which may result in a higher approximation rate by ReLU networks, such as Barron class (Barron, 1993), polynomials (Liang & Srikant, 2016), piecewise smooth function class (Petersen & Voigtlaender, 2018), analytic functions (Wang et al., 2018; Bonito et al., 2021; Schwab & Zech, 2021), Korobov space (Montanelli & Du, 2019; Blanchard & Bennouna, 2021), bandlimited function class (Montanelli et al., 2019). More recently, researchers have broken the lower bound of the approximation rate limited by the metric entropy. They find a much higher approximation rate by networks with other activations or the activations newly designed. On the approximation of Lipschitz functions, some find the exponential approximation rate in terms of depth achieved by FNN with ReLU-sine activation (Yarotsky & Zhevnerchuk, 2020), Floor-ReLU activation (Shen et al., 2020), ReLU-sine-2^x activation (Jiao et al., 2023). Even shallow networks of three layers can achieve the exponential approximation rate in terms of width if the activation is Floor, 2^x, and Step function (Shen et al., 2021). More surprisingly, to approximate continuous function over $[0,1]^d$, (Yarotsky, 2021; Shen et al., 2022a) design the new activation such that FNNs with fixed size can achieve the approximation rate $\mathcal{O}(\epsilon)$ where ϵ is an arbitrarily small constant. As we discussed previously, these models transcend the limit of the metric entropy, hence they are discontinuous approximators, which are unstable in relation to target functions, leading to failure in practice. Nevertheless, from a theoretical standpoint, they are beginning to shatter so-called curse of dimensionality.

3. Theoretical Results and Proof Ideas

We denote by $C\left([0,1]^d;\mathbb{R}^v\right)$ the set of continuous mappings from $[0,1]^d$ to $\mathbb{R}^v$, i.e., and by $L^p\left([0,1]^d\right)$ the set of L^p-integrable functions from $[0,1]^d$ to $\mathbb{R}^v$. Let $\mathcal{T}$ be a compact set of $\mathbb{R}^d$. Define $\omega_f(\cdot)$ is the modulus of continuity of $\boldsymbol{f} = (f_1, \cdots, f_v) \in C\left(\mathcal{T};\mathbb{R}^v\right)$ defined by

$$\omega_f^{\mathcal{T}}(r) := \sup\left\{\mathrm{dist}(\boldsymbol{f}(\mathbf{x}),\boldsymbol{f}(\mathbf{y})) : \mathbf{x},\mathbf{y}\in\mathcal{T}, \|\mathbf{x}-\mathbf{y}\|_2 \leq r\right\},$$

for any $r \geq 0$ where

$$\mathrm{dist}\left(\boldsymbol{f}(\mathbf{x}),\boldsymbol{f}(\mathbf{y})\right) = \max_{1\leq i\leq v} |f_i(\mathbf{x}) - f_i(\mathbf{y})|.$$

Moreover, we define

$$\mathrm{diam}(\mathcal{T}) = \sup\{\|\mathbf{x}-\mathbf{y}\|_\infty : \mathbf{x},\mathbf{y}\in\mathcal{T}\}.$$

The following is our main theorem which extends Theorem 1.1 to the approximation of mappings (vector-valued functions).

Theorem 3.1. *Let $d, v \in \mathbb{N}^+$ and $\mathcal{T}$ be a compact set over $\mathbb{R}^d$. Then for any continuous mapping $\boldsymbol{f} \in C(\mathcal{T};\mathbb{R}^v)$, we have the following approximation results:*
(i) for $p \in [1,\infty)$, there exists a ReLU neural network $\boldsymbol{\rho} : \mathbb{R}^d \to \mathbb{R}^v$ with width $\max\{d+1, v+6\}$ and depth not more than $(4+21v)L + 7d + 8v$ such that

$$\|\boldsymbol{f} - \boldsymbol{\rho}\|_{L^p(\mathcal{T})} \leq C_1(d,p)\cdot\omega_f^{\mathcal{T}}\left(\mathrm{diam}\left(\mathcal{T}\right)L^{-\frac{2}{d}}\right),$$

where $C_1 = 7\sqrt{d}\cdot\left(\mathrm{diam}\left(\mathcal{T}\right)\right)^{\frac{d}{p}} v^{\frac{1}{p}}$.
(ii) for $p = \infty$, there exists a ReLU neural network $\rho : \mathbb{R}^d \to \mathbb{R}^v$ with width $d+v+10$ and depth not more than $6dv3^dL$ such that

$$\|\boldsymbol{f} - \boldsymbol{\rho}\|_{L^\infty(\mathcal{T})} \leq C_2(d)\cdot\omega_f^{\mathcal{T}}\left(\mathrm{diam}\left(\mathcal{T}\right)L^{-\frac{2}{d+1}}\right)$$

where $C_2 = 3^d(3d+1)(6d^{\frac{3}{2}}+1)$.

Remark. The results in the theorem can apply to Lipschitz/Hölder directly according to the definition of ω_f. That is, if f is a μ-Lipschitz continuous function, then $\omega_f(t) \leq \mu t$.

Further, (Zhang et al., 2023) recently investigates the relationship between the ReLU function and diverse ReLU variant activation functions. With the conclusion, it follows that the approximation rate in Theorem 3.1 (or Theorem 1.1) can also be achieved by FNN with various as shown in Corollary 3.2. We list some of the commonly used activation functions in the set $\Sigma = \{$ LeakyReLU, ReLU^2, ELU, SELU, Softplus, Mish, Swish, SiLU, Sigmoid, Tanh, Softsign, dSiLU, SRS, Arctan$\}$. The definition of these activation functions can be found in Appendix D.1.

Corollary 3.2. *Assume a narrow ReLU network with fixed width N_0 and depth $\mathcal{O}(L)$ can achieve the rate as shown in Theorem 1.1 (or Theorem 3.1) for approximating continuous functions. Then narrow networks of width N and depth $\mathcal{O}(L)$ equipped with activation $\sigma \in \Sigma$ can also achieve the same rate. Moreover, the width N satisfies the following proposition.*

*(i) If $\sigma \in \{$*ELU, SELU, Softplus, Mish, Swish, SiLU, $\varrho_1(x), \varrho_2(x), \varrho_3(x)\}$, $N = N_0$. *Here*

$$\begin{aligned}\varrho_1(x) &= x \cdot \mathrm{SiLU}(x),\\ \varrho_2(x) &= x \cdot (\frac{\mathrm{Softsign}(x)}{2} + \frac{1}{2}),\\ \varrho_2(x) &= x \cdot (\frac{\mathrm{Arctan}(x)}{\pi} + \frac{1}{2}).\end{aligned}$$

(ii) If σ is LeakyReLU, $N = 2N_0$.
(iii) If $\sigma \in \{$ ReLU^2, Sigmoid, Tanh, Softsign, dSiLU, SRS$\}$, $N = 3N_0$.

The proof details of Corollary 3.2 can be found in Appendix D.2. Next, we outline the proof idea of Theorem 1.1 and it is natural to be extended to Theorem 3.1.

3.1. Proof Ideas: L^p Norm

In this section, we outline the proof idea of (i) of Theorem 1.1. For a continuous function $f \in [0,1]^d$, we aims to construct a narrow ReLU network $\rho : \mathbb{R}^d \to \mathbb{R}$ with width $d+1$ and depth $\mathcal{O}(L)$, such that $\|\rho - f\|_{L^p([0,1]^d)} = \mathcal{O}(\omega_f(L^{-2/d}))$. By carving out a small region Ω with a small enough Lebesgue measure, and ensuring the approximation rate holds on $[0,1]^d \setminus \Omega$, our objective is met. It is known that step functions are dense in L^p space. Thus, the basic idea is to construct ReLU networks to generate step functions with more steps outside the small region, inspired by the work (Shen et al., 2019a; Lu et al., 2021; Shen et al., 2022b).

Step 1. Space Partitions.

We first divide $[0,1]^d$ into a union of main cubes $\{Q_{\boldsymbol{\beta}}\}$ index by $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ and a trifling region Ω, where K is a proper integer to be determined later.

$$Q_{\boldsymbol{\beta}} := \Big\{\mathbf{x} = [x_1,\cdots,x_d]^T \in [0,1]^d : \\ x_i \in \left[\frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \le K-2\}}\right], i = 1,\cdots,d\Big\}.$$

Moreover, there is a representative $\mathbf{x}_{\boldsymbol{\beta}} \in Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Concretely, $\mathbf{x}_{\boldsymbol{\beta}}$ is the vertex of the cube $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm, i.e., $x_{\boldsymbol{\beta}} = \boldsymbol{\beta}/K$.

Step 2. Mapping $\mathbf{x} \in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$ and encoding it as a scalar.

To achieve this, we begin by examining a one-dimensional scenario. In the one-dimension case, $Q_{\boldsymbol{\beta}}$ for $\beta = k$ is the interval $\left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\right]$. Thus, the following proposition facilitates this task:

Proposition 3.3. *For any $L, d \in \mathbb{N}^+$ and $\delta \in \left(0, \frac{1}{3K}\right]$ with $K = \lfloor L^{2/d} \rfloor$, there exists a function $\hat{\zeta} : [0,1] \to \mathbb{R}^2, \hat{\zeta}(x) = (x, \zeta(x))$ implemented by a ReLU network with width 2 and depth not more than $4L^{\frac{1}{d}} + 3$ such that*

$$\zeta(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\right]$$

for $k = 0, 1, \cdots, K-1$.

The proof of Proposition 3.3 is deferred to Appendix B.5. Then for the multivariate situation, we could construct a mapping $\Phi(\mathbf{x}) = \boldsymbol{\beta}$ for $\mathbf{x} \in Q_{\boldsymbol{\beta}}$ by

$$\Phi(\mathbf{x}) = (\zeta(x_1), \cdots, \zeta(x_d)).$$

With certain techniques, we can implement Φ using a ReLU network of width $d+1$. Note that Proposition 3.3 uses a ReLU network with depth $\mathcal{O}(L^{1/d})$ to realize Φ for $K = \mathcal{O}(L^{2/d})$ many cubes which ensure the optimal approximation rate as detailed in proof details. We end Step 2 by decoding $\boldsymbol{\beta}$ by a scalar:

$$\psi(\boldsymbol{\beta}) := \frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i}.$$

Remark. Note that in this step, d bits of precision are indispensable in the construction. This precision plays a pivotal role in Step 3, where it underpins the conditions of Prop. 3.4 during the point-fitting process. Proposition 3.4 predicates on data with a small variance, i.e., $|y_i - y_j| < \epsilon$. This critical assumption might be violated when precision is finite.

Step 3. Mapping $\psi(\boldsymbol{\beta})$ approximately to $f(\mathbf{x}_{\boldsymbol{\beta}})$.

Note that $\left\{\frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i}\right\}_{\boldsymbol{\beta}} \subset \left\{\frac{j}{2K^d}\right\}_{j=1,2,\cdots,2K^d}$. According to (Shen et al., 2019a; 2022b), there is a function

$g : \mathbb{R} \to \mathbb{R}$ such that $g \circ \psi(\boldsymbol{\beta}) = f(\boldsymbol{\beta})$ and

$$\left| g\left(\frac{j}{2K^d}\right) - g\left(\frac{j-1}{2K^d}\right) \right| \leq \omega_f\left(\frac{\sqrt{d}}{K}\right),$$

for $j = 1, 2, \cdots, 2K^d$. This reduces the problem of approximating f to a point-fitting problem. We need the following proposition where the proof is deferred to B.6.

Proposition 3.4. *Given any $\varepsilon > 0$ and arbitrary $L, J \in \mathbb{N}^+$ with $J \leq L^2$, assuming $y_j \geq 0$ for $j = 0, 1, \cdots, J-1$ satisfy*

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

then there exists a ReLU network $\gamma : \mathbb{R} \to \mathbb{R}$ with width 5 and depth $14L + 8$ such that
(i) $|\gamma(j) - y_j| \leq \varepsilon$ for $j = 0, 1, \cdots, J-1$, and
(ii) $0 \leq \gamma(x) \leq \max_{j=0,1,\cdots,J-1}\{y_j\}$ for any $x \in \mathbb{R}$.

This proposition enables a narrow ReLU network $\phi = \gamma \circ \psi : \mathbb{R}^d \to \mathbb{R}$ such that $\phi(\boldsymbol{\beta}) \approx f(\mathbf{x}_{\boldsymbol{\beta}})$. Crucially, Proposition 3.4 leverages a narrow network with depth $\mathcal{O}(L)$ to solve a point-fitting problem with $\mathcal{O}(L^2)$ points which is central to achieving the optimal approximation rate.

Step 4. Estimation of the network size and the approximation error.

Based on previous steps, we can construct a network $\rho = \phi \circ \Phi$ such that $\rho(\mathbf{x}) = \phi \circ \Phi(\mathbf{x}) = \phi(\boldsymbol{\beta}) \approx f(\mathbf{x}_{\boldsymbol{\beta}}) \approx f(\mathbf{x})$ outside the small region Ω. The last '$\approx$' is achieved by choosing large K, i.e., dividing $[0,1]^d$ into many small enough cubes. Due to the approximation rate being under the L^p norm, it can be achieved in the entire domain $[0,1]^d$. The width of the network ρ is $\max\{d+1, 5\}$ according to Step 2 and 3. If d is not too small, the width $d+1$ is almost the minimum width to satisfy the universality (Hanin & Sellke, 2017; Park et al., 2020; Kim et al., 2023). However, our result guarantees that this minimally wide network can achieve the optimal approximation rate. The proof details can be found in Appendix B.

As a closing remark, Lemma 3.4 (Lu et al., 2021) shows that the approximation rate can be extended from $[0,1]^d/\Omega$ to the entire cube $[0,1]^d$ also under **uniform norm**. However, the width of the network will expand according to their approaches. Hence, under the uniform norm, we will draw another constructive approximation method (Yarotsky, 2018; Yarotsky & Zhevnerchuk, 2020).

3.2. Proof Ideas: Uniform Norm

In this section, we outline the proof of (ii) of Theorem 1.1. We follow the work of (Yarotsky, 2018) to construct a narrow ReLU network with depth $\mathcal{O}(L)$ that achieves the optimal approximation rate. The main challenge is to make its width close to d. At first, let us consider the linear interpolation $\tilde{f}_1$ on the grid $(\mathbb{Z}/n)^d$ with $n \sim L^{1/d}$. If we use $\mathcal{O}(n)$ parameters to construct a sub-network to implement the approximation on a small cube, the number of the total weights of the network to approximate f is $n^d \sim \mathcal{O}(L)$. Then the approximation rate $\|f - \tilde{f}_1\|_\infty = \mathcal{O}\left(\omega_f\left(L^{-1/d}\right)\right)$. To achieve a higher approximation rate, it is worthwhile to consider the refined approximation of $f_2 = f - \tilde{f}_1$ on a smaller grid $(\mathbb{Z}/m)^d$ with $m \sim L^{2/d}$. We expect an approximation rate $\|\tilde{f}_2 - f_2\|_\infty = \mathcal{O}\left(\omega_f\left(L^{-2/d}\right)\right)$ while not significantly expanding the parameter budget beyond $\mathcal{O}(L)$. Thus, we need to consider the linear interpolation approximation $\tilde{f}_2$ of f_2 on a scale $1/m$ and construct a narrow ReLU network to generate $\tilde{f}_2$ with $\mathcal{O}(L)$ parameters. Given that $n^d = \mathcal{O}(L)$ and the function $\tilde{f}_2$ has $\mathcal{O}(m^d)$ information, each cube on a scale $1/n$ contains $(n/m)^d$ information of $\tilde{f}_2$. Hence, in each cube on a scale $1/n$, we need to use $\mathcal{O}(1)$ parameters of a sub-network to encode about $\mathcal{O}\left((m/n)^d\right)$ information. In summary, we will consume about $n^d = \mathcal{O}(L)$ parameters to construct $\tilde{f}_2$ which matches the budget and totally we recover $\mathcal{O}\left((m/n)^d \cdot n^d\right) = \mathcal{O}\left(m^d\right)$ information.

The key of the approximation of $\tilde{f}_2$ on a refined scale $1/m$ without increasing the number of parameters is to use $\mathcal{O}(1)$ parameters to encode about $\mathcal{O}\left((m/n)^d\right)$ information and decode it. This process can be realized by the bit-extraction technique, which has been widely used in modern constructive methods (Shen et al., 2022b; Yarotsky & Zhevnerchuk, 2020). Recall that the domain is partitioned into a union of small cubes $Q_{\mathbf{k}}$ on a scale $1/n$. In this key process, it is essential to determine which specific small cube the input vector $\mathbf{x}$ falls into. Thus, similar to the function Φ in section 3.1, we also have a mapping $\Psi : \mathbb{R}^d \to \mathbb{R}^d$ such that $\Psi(\mathbf{x}) = (\psi_1(x_1), \cdots, \psi_d(x_d)) = \mathbf{k}/n$ if $\mathbf{x}$ belongs to the small cube $Q_{\mathbf{k}}$. In (Yarotsky, 2018), the author uses d neurons in each layer to forward the value of the input $\mathbf{x}$ and another d neurons to forward the value of $\Psi(\mathbf{x})$. With some extra neurons in each layer to do the intermediate computation, they cost $2d + \mathcal{O}(1)$ neurons in each layer. However, in our paper, to make the network narrow as much as possible, instead of using neurons to store the value of $\Psi(\mathbf{x})$ we compute them as the intermediate computation in each constructive stage. The cost is that we get a little lower rate $\mathcal{O}\left(L^{-2/(d+1)}\right)$. But anyway, this approximation rate is almost optimal. The proof details can be found in C.

3.3. Optimality

In this section, we show that the approximation rate $\mathcal{O}(\omega_f(L^{-2/d}))$ is optimal for both L^p norm and uniform norm. Specifically, we will see that there is no room to improve this rate on the approximation of Lipschitz functions over $[0,1]^d$. We denoted by $\mathrm{Lip}([0,1]^d; \mu > 0)$ the

set of Lipschitz functions over $[0,1]^d$ and for any $f \in \mathrm{Lip}([0,1]^d; \mu > 0)$, we have $|f(\mathbf{x}) - f(\mathbf{y})| \leq \mu \|\mathbf{x} - \mathbf{y}\|$.

As we mentioned before, the rate is lower bounded by the so-called VC dimension of the hypothesis space. Thus, we first give the definition. Let $\mathcal{H}$ be a function family consisting of functions from $\mathcal{X} \subset \mathbb{R}^d$ to $\{0,1\}$. For any non-negative integer m, the growth function of $\mathcal{H}$ is defined as

$$\Pi_H(m) := \max_{\mathbf{x}_1,\ldots,\mathbf{x}_m \in \mathcal{X}} \left|\{(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)) : h \in \mathcal{H}\}\right|.$$

If $|\{(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)) : h \in \mathcal{H}\}| = 2^m$, we say $\mathcal{H}$ shatters the set $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$. Moreover, let $\mathscr{F}$ be a set of functions from $\mathcal{X}$ to $\mathbb{R}$, we say $\mathscr{F}$ shatters the set $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ if $S \circ \mathscr{F}$ shatters the set $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ where

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

The VC dimension of $\mathcal{H}$, denoted by $\mathrm{VCdim}(\mathcal{H})$, is the size of the largest shattered set, i.e. the largest m such that $\Pi_H(m) = 2^m$. If there is no largest m, we define $\mathrm{VCdim}(\mathcal{H}) = \infty$.

Theorem 3.5. *Let $p \in [1,\infty]$, $d \in \mathbb{N}^+$ and $\mathcal{H}_\sigma(L)$ be the set of FNNs of fixed width and depth $\mathcal{O}(L)$ with some activation function σ. Define*

$$\mathcal{E}(d,L) := \sup_{f \in \mathrm{Lip}([0,1]^d,\mu)} \left(\inf_{\rho \in \mathcal{H}_\sigma(L)} \|\rho - f\|_{L^p([0,1]^d)} \right).$$

If $VCdim(\mathcal{H}_\sigma(L)) \leq D$, then we have

$$\mathcal{E}(d,L) \geq C(p,d) D^{-\frac{1}{d}}$$

where $C(p,d)$ is a constant may depend on p and d.

For $p = \infty$, this result is the direct corollary of (Yarotsky, 2017; 2018; Shen et al., 2019a; 2022b) and recently, (Siegel, 2023) prove it for $p \in [1,\infty)$. The proof details can be found in Appendix D.3.

Corollary 3.6. *With the same notation as Theorem 3.5, we have the following result for various activation functions σ.*

(i) If σ belongs to $\{$piecewise polynomial (e.g. ReLU, ReLU^2, $\mathrm{LeakyReLU}$*), Softsign$\}$, we have $\mathrm{VCdim}(\mathcal{H}_\sigma(L)) = \mathcal{O}(L^2)$. Thus,*

$$\mathcal{E}(d,L) \geq C(p,d) L^{-\frac{2}{d}}.$$

(ii) If σ belongs to $\{\mathrm{ELU}, \mathrm{SELU}, \mathrm{SiLU}, \mathrm{Swish}, \mathrm{Mish}, \mathrm{Sigmoid}, \mathrm{Tanh}, \mathrm{dSiLU}, \mathrm{SRS}, \mathrm{Arctan}\}$, we have $\mathrm{VCdim}(\mathcal{H}_\sigma(L)) = \mathcal{O}(L^4)$. Thus, for these activation functions, we have

$$\mathcal{E}(d,L) \geq C(p,d) L^{-\frac{4}{d}}.$$

In Corollary 3.6, (i) is the result of Theorem 8.4 (Anthony et al., 1999) or (Bartlett et al., 2019) and (ii) is the result of (Karpinski & Macintyre, 1997) or Theorem 8.14 (Anthony et al., 1999). We have the details of the explanation in D.4. Besides, it remains open whether the bound (ii) of Corollary 3.6 can be improved.

It follows from Corollary 3.6 that the rate $\mathcal{O}(L^{-2/d})$ is optimal for piecewise polynomials activation such as LeakyReLU, ReLU^2 and for Softsign. For other activation functions in (ii) of Corollary 3.6, the rate achieved by narrow networks may be suboptimal but higher than the previous results. For those activation functions in (i) of Corollary 3.2, they keep the minimum width to achieve the higher approximation rate. Moreover, we can see that an activation $\varrho_2(x)$ simultaneously keeps the minimum width and achieves the optimal approximation rate.

4. Concluding Remarks and Discussion

Non-compact Domains. While some readers may be curious about how will the result change if the domain is not compact, we provide some explanations here from two aspects. First, for uniform approximation, the results generally do not hold on non-compact domains. For example, we take the task of approximating the continuous function $f(x) = \frac{1}{x}$ on the bounded yet open interval $(0,1]$, which is not compact. ReLU neural networks g produce continuous piecewise linear functions and are thus bounded on $[0,1]$. However, for any desired error ϵ, the difference $|f(x) - g(x)|$ will inevitably exceed ϵ in the vicinity of 0, due to the unbounded behavior of f. However, the situation is different for L^p approximation, i.e., the results will still hold on a (measurable) non-compact set. The explanation is in the following. The space of continuous functions with compact support is dense in $L^p(\mathbb{R}^d)$ (Walter, 1987), meaning that for any $f \in L^p(\mathbb{R}^d)$ and any error $\epsilon > 0$, there exists a continuous function h with compact support such that $\|f - h\|_{L^p} \leq \epsilon$. Therefore, when the domain of interest is $\mathbb{R}^d$, one can effectively reduce the problem to that of a compact domain for L^p approximation. Furthermore, if the target function f is initially defined over a measurable subset $E \subset \mathbb{R}^d$, it can be extended to $\mathbb{R}^d$ by setting $f(x) = 0$ for $x \in \mathbb{R}^d \setminus E$, thereby allowing for a reduction to a compact domain scenario.

Non-trivial Extension. It has been shown that any ReLU FNN of width N and depth L can be approximated by a narrow network width $d+2$ and depth $\mathcal{O}(L^2)$ (Vardi et al., 2022). However, if we directly apply this conclusion to (Shen et al., 2019a; Yarotsky, 2018; Shen et al., 2022b), the rate would become suboptimal because it just uses a narrow network of depth $\mathcal{O}(L^2)$ to achieve the rate $\mathcal{O}(L^{-2/d})$ for approximating Lipschitz functions. Thus, our work showing that ReLU FNN with the minimal width can achieve the

optimal approximation rate is a non-trivial extension of previous work.

Approximation of Mappings from $[0,1]^d$ to $\mathbb{R}^v$. As shown in (Kim et al., 2023), the minimal width of ReLU FNNs to satisfy the universal approximation property for Lebesgue-integrable mappings from $[0,1]^d$ to $\mathbb{R}^v$ is exactly $\max\{d, v, 2\}$. Our result reveals that a ReLU FNN with a width $\max\{d+1, v+6\}$ slightly greater than the minimum by a small constant, satisfies not only universality but also ensures an optimal approximation rate. Furthermore, our work shows that for continuous mappings from $[0,1]^d$ to $\mathbb{R}^v$, ReLU networks with a width of $d+v+10$ are capable of achieving nearly optimal approximation. Meanwhile, the upper bound of the minimum width of ReLU networks for uniform universal approximation remains $d+v+1$. Our work shows a width slightly exceeding this by an absolute constant can still guarantee the optimal approximation for continuous mappings over a compact domain.

Approximation Rate of Other Neural Network Structures. Recently, approximation capabilities have also been widely studied for various neural network architectures such as convolutional neural networks (Zhou, 2018; 2020b) and ResNet (Lin & Jegelka, 2018; Oono & Suzuki, 2019). It has been shown (Zhou, 2020a) that any FNN could be represented by a downsampled CNN with the same order number of parameters. Thus, it follows that ReLU CNN can also achieve the optimal approximation rate for continuous functions in terms of width and depth. Besides, the construction of FNN is also possible for ResNet which allows them to achieve the optimal approximation for continuous functions.

Diverse Activation Functions. Our work discusses the approximation capabilities of FNNs with various activation functions. FNNs with different activation functions may have different VC dimensions, hence resulting in different approximation capabilities. Narrow networks with piecewise polynomial activation functions, such as ReLU and Leaky-ReLU, share the same order VC dimension $\mathcal{O}(L^2)$. While (Cai, 2022; Kim et al., 2023; Duan et al., 2023) show that LeakyReLU networks with width d can achieve universal approximation for Lebesgue-integrable functions over a compact domain in $\mathbb{R}^d$, we use a network with a larger width, $2d+\mathcal{O}(1)$ to guarantee the optimality. However, the approximation rate of the minimally wide LeakyReLU networks in (Cai, 2022; Kim et al., 2023) is implicit. One might need a large number of parameters while using a LeakyReLU network with width d for approximation. Moreover, it is a non-trivial and open question whether a LeakyReLU network width $d+\mathcal{O}(1)$ can achieve the optimal approximation for Lebesgue-integrable functions over a compact domain in $\mathbb{R}^d$.

Depth-Width Trade-offs. This paper focuses on the approximation rates of narrow networks, specifically how these rates correlate with network depth. However, readers may also be interested in exploring the depth-width trade-offs in network architectures within approximation theory. This subject has received extensive attention in the previous literature, with key findings illustrated in Fig. 1. It is important to note that a shallow network with a width of N typically has around $\mathcal{O}(N^2)$ parameters. Therefore, the approximation rate of $\mathcal{O}(N^{-2/d})$ with respect to width becomes $\mathcal{O}(W^{-1/d})$ when expressed in terms of the total number of parameters W. When compared to the optimal rate of $\mathcal{O}(L^{-2/d})$ associated with a depth of $\mathcal{O}(L)$, it becomes evident that deep networks tend to be more parameter-efficient in function approximation due to the number of parameters $W=\mathcal{O}(L)$.

Heavy Dependency on the Number of Parameters. Readers might note that the approximation rates heavily depend on the input dimension d, width N, and depth L. We remark on this briefly. In approximation theory, the approximation rate of ReLU networks is often expressed as $\mathcal{O}(c(d)N^{-\frac{2}{d}}L^{-\frac{2}{d}})$ when approximating a Lipschitz continuous function. The rate's dependency on N and L is intrinsic, tied to the limitations of the network's VC dimension as pointed out in Sec. 2. Besides, the rate heavily depends on d because of the worst-case analysis in the proof, and improving this dependency remains an open challenge.

Acknowledgements

This work is supported in part by a General Research Fund from Research Grants Council, Hong Kong (Project No. 11203122), an InnoHK initiative, The Government of the HKSAR, Laboratory for AI-Powered Financial Technologies, and a Shenzhen-Hong Kong-Macau Science & Technology Project (Category C, Project No. SGDX20220530111203026). The authors would also like to thank the anonymous reviewers for their helpful comments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Anthony, M., Bartlett, P. L., Bartlett, P. L., et al. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*, 20(1):2285–2301, 2019.

Blanchard, M. and Bennouna, M. A. Shallow and deep networks are near-optimal approximators of korobov functions. In *International Conference on Learning Representations*, 2021.

Bonito, A., DeVore, R., Guignard, D., Jantsch, P., and Petrova, G. Polynomial approximation of anisotropic analytic functions of several variables. *Constructive Approximation*, 53(2):319–348, 2021.

Cai, Y. Achieve the minimum width of neural networks for universal approximation. In *The Eleventh International Conference on Learning Representations*, 2022.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Daubechies, I., DeVore, R., Foucart, S., Hanin, B., and Petrova, G. Nonlinear approximation and (deep) relu networks. *Constructive Approximation*, 55(1):127–172, 2022.

DeVore, R. A., Howard, R., and Micchelli, C. Optimal nonlinear approximation. *Manuscripta mathematica*, 63: 469–478, 1989.

Duan, Y., Ji, G., Cai, Y., et al. Minimum width of leaky-relu neural networks for uniform universal approximation. In *International Conference on Machine Learning*, pp. 19460–19470. PMLR, 2023.

Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. In *Conference on learning theory*, pp. 907–940. PMLR, 2016.

Geuchen, P., Jahn, T., and Matt, H. Universal approximation with complex-valued deep narrow neural networks. *arXiv preprint arXiv:2305.16910*, 2023.

Goldberg, P. and Jerrum, M. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 361–369, 1993.

Hanin, B. and Sellke, M. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

He, J., Li, L., and Xu, J. Approximation properties of deep relu cnns. *Research in the Mathematical Sciences*, 9(3): 38, 2022.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Jiao, Y., Lai, Y., Lu, X., Wang, F., Yang, J. Z., and Yang, Y. Deep neural networks with relu-sine-exponential activations break curse of dimensionality in approximation on hölder class. *SIAM Journal on Mathematical Analysis*, 55(4):3635–3649, 2023.

Karpinski, M. and Macintyre, A. Polynomial bounds for vc dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54 (1):169–176, 1997.

Kidger, P. and Lyons, T. Universal approximation with deep narrow networks. In *Conference on learning theory*, pp. 2306–2327. PMLR, 2020.

Kim, N., Min, C., and Park, S. Minimum width for universal approximation using relu networks on compact domain. *arXiv preprint arXiv:2309.10402*, 2023.

Kolmogorov, A. N. and Tikhomirov, V. M. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

Kratsios, A. and Papon, L. Universal approximation theorems for differentiable geometric deep learning. *The Journal of Machine Learning Research*, 23(1):8896–8968, 2022.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Li, Q., Lin, T., and Shen, Z. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 2022a.

Li, Z., Han, J., Li, Q., et al. On the curse of memory in recurrent neural networks: Approximation and optimization analysis. *arXiv preprint arXiv:2009.07799*, 2020.

Li, Z., Han, J., Weinan, E., and Li, Q. Approximation and optimization theory for linear continuous-time recurrent neural networks. *J. Mach. Learn. Res.*, 23:42–1, 2022b.

Liang, S. and Srikant, R. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*, 2016.

Lin, H. and Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. *Advances in neural information processing systems*, 31, 2018.

Lu, J., Shen, Z., Yang, H., and Zhang, S. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

McShane, E. J. Extension of range of functions. 1934.

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

Montanelli, H., Yang, H., and Du, Q. Deep relu networks overcome the curse of dimensionality for bandlimited functions. *arXiv preprint arXiv:1903.00735*, 2019.

Oono, K. and Suzuki, T. Approximation and non-parametric estimation of resnet-type convolutional neural networks. In *International conference on machine learning*, pp. 4922–4931. PMLR, 2019.

Park, S., Yun, C., Lee, J., and Shin, J. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2020.

Petersen, P. and Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

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

Sannai, A., Takai, Y., and Cordonnier, M. Universal approximations of permutation invariant/equivariant functions by deep neural networks. *arXiv preprint arXiv:1903.01939*, 2019.

Schwab, C. and Zech, J. Deep learning in high dimension: Neural network approximation of analytic functions in $L^2(\mathbb{R}^d, \gamma_d)$. *arXiv preprint arXiv:2111.07080*, 2021.

Segol, N. and Lipman, Y. On universal equivariant set networks. In *International Conference on Learning Representations*, 2019.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation characterized by number of neurons. *arXiv preprint arXiv:1906.05497*, 2019a.

Shen, Z., Yang, H., and Zhang, S. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019b.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation with discrepancy being reciprocal of width to power of depth. *arXiv preprint arXiv:2006.12231*, 2020.

Shen, Z., Yang, H., and Zhang, S. Neural network approximation: Three hidden layers are enough. *Neural Networks*, 141:160–173, 2021.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons. *The Journal of Machine Learning Research*, 23(1):12653–12712, 2022a.

Shen, Z., Yang, H., and Zhang, S. Optimal approximation rate of relu networks in terms of width and depth. *Journal de Mathématiques Pures et Appliquées*, 157:101–135, 2022b.

Siegel, J. W. Optimal approximation rates for deep relu neural networks on sobolev and besov spaces. *Journal of Machine Learning Research*, 24(357):1–52, 2023.

Telgarsky, M. Benefits of depth in neural networks. In *Conference on learning theory*, pp. 1517–1539. PMLR, 2016.

Vardi, G., Yehudai, G., and Shamir, O. Width is less important than depth in relu neural networks. In *Conference on Learning Theory*, pp. 1249–1281. PMLR, 2022.

Voigtlaender, F. The universal approximation theorem for complex-valued neural networks. *Applied and Computational Harmonic Analysis*, 64:33–61, 2023.

Walter, R. Real and complex analysis. 1987.

Wang, Q. et al. Exponential convergence of the deep neural network approximation for analytic functions. *arXiv preprint arXiv:1807.00297*, 2018.

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

Yarotsky, D. Optimal approximation of continuous functions by very deep relu networks. In *Conference on learning theory*, pp. 639–649. PMLR, 2018.

Yarotsky, D. Elementary superexpressive activations. In *International Conference on Machine Learning*, pp. 11932–11940. PMLR, 2021.

Yarotsky, D. and Zhevnerchuk, A. The phase diagram of approximation rates for deep neural networks. *Advances in neural information processing systems*, 33:13005–13015, 2020.

Zhang, S., Lu, J., and Zhao, H. Deep network approximation: Beyond relu to diverse activation functions. *arXiv preprint arXiv:2307.06555*, 2023.

Zhou, D.-X. Deep distributed convolutional neural networks: Universality. *Analysis and Applications*, 16(06):895–919, 2018.

Zhou, D.-X. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020a.

Zhou, D.-X. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48 (2):787–794, 2020b.

A. Prelimineries and Notations

A.1. Notations

We summarize the notations we will use in this paper in the following.

- We denote by $\mathbb{R}$ the set of real numbers, by $\mathbb{N}$ the set of natural numbers $0, 1, 2, \cdots$, by $\mathbb{Z}$ the set of integers. Moreover, $\mathbb{N}^+ := \mathbb{N}/\{0\}$.
- We use non-bold letters like x, y, z for scalars, and boldface letters like $\mathbf{x}, \mathbf{y}, \mathbf{z}$ for vectors in the Euclidean space. Moreover, we use normal, non-bold letters like f, g for scalar-valued functions, shortened as functions, and normal bold letters like $\boldsymbol{f}, \boldsymbol{g}$ for vector-valued functions, shortened as mappings.
- For any $p \in [1, \infty)$, the p-norm of a vector $\mathbf{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$ is defined by

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

- Denote by $\mu(\mathcal{T})$ the Lebesgue measure of a measurable set $\mathcal{T}$.
- Let $\mathbb{1}_S$ be the characteristic function on a set S, i.e., $\mathbb{1}_S = 1$ on set S and 0 otherwise.
- For two sets A, B, $A \backslash B := \{x : x \in A, x \notin B\}$.
- For any $\xi \in \mathbb{R}$, let $\lfloor \xi \rfloor := \max\{i : i \leq \xi, i \in \mathbb{Z}\}$ and $\lceil \xi \rceil := \min\{i : i \geq \xi, i \in \mathbb{Z}\}$.
- Let $d, v \in \mathbb{N}^+$ and $\mathcal{T}$ be some compact (and measurable) set of $\mathbb{R}^d$.

  Then we denote by $C(\mathcal{T})$ the set of continuous functions from $\mathcal{T}$ to $\mathbb{R}$ with the norm $\|f(x)\|_{L^\infty(\mathcal{T})} = \sup_{x \in \mathcal{T}} |f(x)|$. Similarly, for $p \in [1, \infty)$, $L^p(\mathcal{T})$ denotes the space of p-integrable measurable functions on $\mathcal{T}$, with norm $\|f\|_{L^p(\mathcal{T})} = \left(\int_E |f(x)|^p dx\right)^{1/p}$.

  Moreover, denote by $C(\mathcal{T}; \mathbb{R}^v)$ the set of continuous mappings from $\mathcal{T}$ to $\mathbb{R}^v$ with the norm

$$\|\boldsymbol{f}\|_{L^\infty(\mathcal{T})} = \max_{1 \leq i \leq v} \sup_{\mathbf{x} \in \mathcal{T}} |f_i(\mathbf{x})|.$$

  Similarly, for $p \in [1, \infty)$, $L^p(\mathcal{T}; \mathbb{R}^v)$ denotes the space of p-integrable measurable mappings from $\mathcal{T}$ to $\mathbb{R}^v$, with the norm

$$\|\boldsymbol{f}\|_{L^p(\mathcal{T})} = \left(\int_E \|\boldsymbol{f}(\mathbf{x})\|_p^p \, \mathrm{d}\mathbf{x}\right)^{1/p} = \left(\int_E \left(\sum_{i=1}^v |f_i(\mathbf{x})|^p\right) \, \mathrm{d}\mathbf{x}\right)^{1/p}.$$

- Assume $\boldsymbol{n} \in \mathbb{N}^d$, then $f(\boldsymbol{n}) = \mathcal{O}(g(\boldsymbol{n}))$, $f(\boldsymbol{n}) = \Theta(g(\boldsymbol{n}))$, $f(\boldsymbol{n}) = \Omega(g(\boldsymbol{n}))$, respectively, implies that there exists positive C, C_1, C_2 independent of $\boldsymbol{n}$, f, and g such that $f(\boldsymbol{n}) \leq Cg(\boldsymbol{n}), C_1 g(\boldsymbol{n}) \leq f(\boldsymbol{n}) \leq C_2 g(\boldsymbol{n}), f(\boldsymbol{n}) \geq Cg(\boldsymbol{n})$ when all entries of $\boldsymbol{n}$ go to $+\infty$.
- Let $\mathcal{T}$ be a compact set of $\mathbb{R}^d$. $\omega_f(\cdot)$ is the modulus of continuity of $\boldsymbol{f} = (f_1, \cdots, f_v) \in C(\mathcal{T}; \mathbb{R}^v)$ defined by

$$\omega_f^{\mathcal{T}}(r) := \sup\left\{\mathrm{dist}(\boldsymbol{f}(\mathbf{x}), \boldsymbol{f}(\mathbf{y})) : \mathbf{x}, \mathbf{y} \in \mathcal{T}, \|\mathbf{x} - \mathbf{y}\|_2 \leq r\right\},$$

  for any $r \geq 0$ where

$$\mathrm{dist}\left(\boldsymbol{f}(\mathbf{x}), \boldsymbol{f}(\mathbf{y})\right) = \begin{cases} |f(\mathbf{x}) - f(\mathbf{y})|, & v = 1, \\ \max_{1 \leq i \leq v} |f_i(\mathbf{x}) - f_i(\mathbf{y})|, & v > 1. \end{cases}$$

  For conciseness, we write $\omega_f(r) := \omega_f^{\mathcal{T}}(r)$ when $\mathcal{T} = [0, 1]^d$.
- For conciseness, we denote $(x_1, x_2, \cdots, x_d) := [x_1, \cdots, x_d]^T \in \mathbb{R}^d$ for $x_1, \cdots, x_d \in \mathbb{R}$ and $(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_d) := [\mathbf{z}_1^T, \mathbf{z}_2^T, \cdots, \mathbf{z}_d^T] \in \mathbb{R}^{m_1 + m_2 + \cdots + m_d}$ where $\mathbf{z}_i \in \mathbb{R}^{m_i}$ for $i = 1, 2, \cdots, d$.

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

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

  In particular, $\Omega\left([0,1]^d, K, \delta\right) = \varnothing$ if $K = 1$.

- For $\theta \in [0,1)$, suppose its base-q representation is $\theta = \sum_{\ell=1}^{\infty} \theta_\ell q^{-\ell}$ with $\theta_\ell \in \{0, 1, \cdots, q-1\}$. Then we use the notation $0.\theta_1\theta_2\cdots\theta_L$ to denote the L-term base-q representation of θ, i.e., $0.\theta_1\theta_2\cdots\theta_L := \sum_{\ell=1}^{L} \theta_\ell q^{-\ell}$.
- Let $\mathcal{T} \subset \mathbb{R}^d$ be a compact (and measurable) set. We define $\mathrm{diam}(\mathcal{T}) = \sup\{\|\mathbf{x} - \mathbf{y}\|_\infty : \mathbf{x}, \mathbf{y} \in \mathcal{T}\}$.
- For a univariate continuous piecewise linear function $f(x)$, x_0 is called a breakpoint of f if $\lim_{x \to x_0^+} f'(x) \neq \lim_{x \to x_0^-} f'(x)$. We will abbreviate 'continuous piecewise linear' as 'CPwL'.
- For a finite sample set $A = \{(x_i, y_i) : 1 \leq i \leq m\}$ and let x_i be increasing for i, we have a CPwL function f such that: i) $f(x_i) = y_i$ and ii) f is linear in each interval $(-\infty, x_1], [x_1, x_2], \cdots, [x_{m-1}, x_m], [x_m, \infty]$. In this case, we say f is a CPwL function defined by the sample set A. Note that the set of the breakpoints of f is a subset of $\{x_1, \cdots, x_m\}$.
- e is the base of the natural logarithm, i.e., $e = \lim_{n \to \infty}(1 + \frac{1}{n})^n$.

Next, we introduce some neural network architecture we will discuss in this paper.

A.2. Feedforward Neural Networks (FNNs)

As is known to all, FNN is a mapping $\boldsymbol{f} : \mathbb{R}^d :\to \mathbb{R}^v$ which is formed as the alternating compositions of an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, and affine transformations $\mathcal{A}^{[i]}(y) = U_i y + v_i$ with $U_i \in \mathbb{R}^{d_i \times d_{i-1}}, v_i \in \mathbb{R}^{d_i}, d_0 = d$ for $i = 1, 2, \cdots, L$. Specifically,

$$\boldsymbol{f}(\mathbf{x}) = \mathcal{L} \circ \sigma \circ \mathcal{A}^{[L]} \circ \sigma \circ \mathcal{A}^{[L-1]} \circ \cdots \circ \sigma \circ \mathcal{A}^{[1]}(\mathbf{x})$$

where $\mathcal{L}$ is a final affine transformation and for $\mathbf{x} \in \mathbb{R}^d$, $\sigma(\mathbf{x}) := (\sigma(x_1), \cdots, \sigma(x_d))$. Here L denotes the number of layers of the FNN, and the width of the FNN is conventionally defined by $\max\{d_1, d_2, \cdots, d_L\} := K$. As usual, we use ReLU as the activation function defined by:

$$\mathrm{ReLU}(x) = \max(x, 0) = (x)_+, x \in \mathbb{R}.$$

Typically, it is presumed that the number of neurons in each layer of an FNN is the same, which is equal to the width K, as any neuron deficits in a layer can be dealt with by adding $K - d_j$ neurons whose biases are zero in layer j. The weights between these extra neurons are consequently assigned to zero.

We denote by $\mathcal{NN}_\sigma^{d,v}(N, L)$ the set of all FNN mappings from some compact set $\mathcal{T} \subset \mathbb{R}^d$ to $\mathbb{R}^v$ with width N, depth L, and activation function σ. In cases without ambiguity, we may omit the subscripts d, v and superscripts σ for conciseness. Moreover, we commonly refer to all neurons from a fixed row as a **channel**.

A.3. Register Models

To describe the construction of networks concisely and conveniently in this paper, we introduce a special network architecture that we call the register model, which designates certain channels for specific tasks. Given an input $\mathbf{x} \in \mathbb{R}^d$, we use the top d channels to reserve the input values. This allows us to use x as an input to the computation performed at any later layer of the network. We refer to these channels **source channels.** Moreover, we will designate the bottom v channels to simply aggregate the value of certain intermediate computations and may push them forward. These channels are called **collation channels**. Note that the neurons in source channels and collation channels are activation-free, i.e., their activation function is $x \mapsto x$. The rest of the channels are used for normal computation in which the neurons are equipped with given activation functions. The specific definition of the register model is given in the following.

Definition A.1. Let $d, v, n, m, C \in \mathbb{N}_+$, $\sigma : \mathbb{R} \to \mathbb{R}$ and $\mathrm{Aff}\left(\mathbb{R}^n; \mathbb{R}^m\right)$ be the set of affine transformations from $\mathbb{R}^n$ to $\mathbb{R}^m$. Denote by $\mathcal{I}_\sigma^{d,v}(d, C, m; L)$ the set of register models and each $h \in \mathcal{I}_\sigma^{d,v}(d, C, m; L)$ is a mapping from some compact set $\mathcal{T} \subset \mathbb{R}^d$ to $\mathbb{R}^v$ defined by

$$T_L \circ \tilde{\sigma} \circ \ldots \circ \tilde{\sigma} \circ T_1,$$

where $T_1 \in \text{Aff}\left(\mathbb{R}^d;\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m\right), T_L \in \text{Aff}\left(\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m;\mathbb{R}^v\right), T_\ell \in \text{Aff}\left(\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m;\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m\right)$ for $\ell\in\{2,\ldots,L-1\}$, and the map $\tilde{\sigma}$ acts on $\mathbb{R}^{d+C+m}$ via

$$\tilde{\sigma}:\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m\to\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m, \tag{2}$$

$$\left(\tilde{\sigma}\left(x_1,\ldots,x_{d+C+m}\right)\right)_j=\begin{cases}\sigma\left(z_j\right), & j\in\{d+1,\cdots,d+C\},\\ z_j, & j\in\{1,2,\cdots,d+C+m\}/\{d+1,\cdots,d+C\}.\end{cases} \tag{3}$$

Let $x\in\mathbb{R}^d$ and $z=(z_1,z_2,z_3)\in\mathbb{R}^d\times\mathbb{R}^C\times\mathbb{R}^m$. T_ℓ for $\ell\in\{1,2,\ldots,L-1\}$ is futher defined by

$$\begin{aligned}&T_1(x)=(x,\mathcal{A}_1(x),\mathcal{B}_1(x)),\quad \text{and}\\ &T_\ell(z_1,z_2,z_3)=(z_1,\mathcal{A}_\ell(z_1,z_2),\mathcal{B}_\ell(z_1,z_2,z_3)),\quad \ell=2,3,\cdots,L-1\end{aligned}$$

where $\mathcal{A}_1\in\text{Aff}\left(\mathbb{R}^d;\mathbb{R}^C\right), \mathcal{B}_1\in\text{Aff}\left(\mathbb{R}^d;\mathbb{R}^m\right)$ and $\mathcal{A}_\ell\in\text{Aff}\left(\mathbb{R}^{d+C};\mathbb{R}^C\right), \mathcal{B}_\ell\in\text{Aff}\left(\mathbb{R}^{d+C+m};\mathbb{R}^m\right)$ for $\ell=2,3,\cdots,L-1$.

If $v=1$, we will simply denote $\mathcal{I}_\sigma^{d,v}(d,C,m;L)$ as $\mathcal{I}_\sigma(d,C,m;L)$. According to the discussion before, in a register model belonging to $\mathcal{I}_\sigma^{d,v}(d,C,m;L)$, the top d channels are called source channels and the bottom m channels are called collation channels. Here the definition of σ action on $\mathbb{R}^{d+C+m}$ (3) just implies the neurons in source channels and collation channels are activation-free.

For a register model in $\mathcal{I}_\sigma^{d,v}(d,C,m;L)$, note that the domain is compact. Then for each source and collation channel, we can always find $C_j(j=1,2,\cdots,L)$ such that $S^{(j)}+C_j\geq 0$ where $S^{(j)}(j=1,2,\cdots,L)$. Hence $S^{(j)}=\text{ReLU}\left(S^{(j)}(x)+C_j\right)-C_j$. With this trick, we have the following conclusion.

Lemma A.2 (Remark 3.1 (Daubechies et al., 2022))**.** *Let $\mathcal{I}_{\text{ReLU}}^{d,v}(d,C,m;L)$ be the set of register models from $\mathcal{T}\subset\mathbb{R}^d$ to $\mathbb{R}^v$. Then $\mathcal{I}_{\text{ReLU}}^{d,v}(d,C,m;L)\subset\mathcal{NN}_{\text{ReLU}}^{d,v}(d+C+m,L)$.*

This lemma states that a ReLU register model is also a ReLU network with the same width and depth.

A.4. Extending Approximation from $[0,1]^d$ to an irregular Domain

In this section, we show how to extend the approximation result from a hypercube $[0,1]^d$ to a compact domain $\mathcal{T}\subset\mathbb{R}^d$. With these results, it suffices to consider the approximation of functions on $[0,1]^d$ later.

Lemma A.3. *Let $N,L\in\mathbb{N}^+$ and $\mathcal{T}\subset\mathbb{R}^d$ be a compact (and measurable) set. Suppose $f:[0,1]^d\to\mathbb{R}$ is a continuous function and for $p\in[1,\infty]$, there exists a ReLU network ρ with width N and depth L such that*

$$\|f-\rho\|_{L^p([0,1]^d)}\leq C_1\omega_f\left(r\right),$$

for some $r>0$ and C_1 is some constant that can depends on d or p. Then for any $g\in C(\mathcal{T})$ and $p\in[1,\infty]$, there is a ReLU network ρ with width N and depth L such that

$$\|g-\rho\|_{L^p(\mathcal{T})}\leq C_1\left(\text{diam}\left(\mathcal{T}\right)\right)^{d/p}\omega_g^{\mathcal{T}}\left(\text{diam}\left(\mathcal{T}\right)r\right).$$

Proof. By assumption, $\mathcal{T}$ is a compact set. Hence, g can be extended to $[-R,R]^d$ (for some $R>0$ and $\mathcal{T}\subset[-R,R]^d$) preserving its modulus of continuity by Theorem 2 in (McShane, 1934). Thus, without loss of generality, we can assume $\mathcal{T}$ is connected. We define

$$\widetilde{g}(\mathbf{x}):=g(\text{diam}(\mathcal{T})\mathbf{x}+\inf\mathcal{T}),\qquad \text{for any }\mathbf{x}\in[0,1]^d.$$

For $\widetilde{g}\in C\left([0,1]^d\right)$, by assumption there exists a ReLU FNN $\widetilde{\rho}$ with width N and depth L such that

$$\|\widetilde{\rho}-\widetilde{g}\|_{L^p([0,1]^d)}\leq C_1\omega_{\widetilde{g}}\left(r\right),$$

Note that $g(\mathbf{x})=\widetilde{g}\left(\frac{\mathbf{x}-\inf\mathcal{T}}{\text{diam}(\mathcal{T})}\right)$ for any $\mathbf{x}\in\mathcal{T}$ and

$$\omega_{\widetilde{g}}(t)=\omega_g^{\mathcal{T}}(\text{diam}(\mathcal{T})t),\qquad \text{for any } t\geq 0.$$

Define $\rho(\mathbf{x}) := \widetilde{\rho}\left(\frac{\mathbf{x}-\inf \mathcal{T}}{\operatorname{diam}(\mathcal{T})}\right) = \widetilde{\rho} \circ \mathcal{L}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^d$, where $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}^d$ is an affine linear map given by $\mathcal{L}(\mathbf{x}) = \frac{\mathbf{x}-\inf \mathcal{T}}{\operatorname{diam}(\mathcal{T})}$. Then ρ is a ReLU network with width N and depth L. Moreover, for any $\mathbf{x} \in \mathcal{T}$, we have $\frac{\mathbf{x}-\inf \mathcal{T}}{\operatorname{diam}(\mathcal{T})} \in [0,1]^d$. Then for $p \in [1,\infty]$ we have

$$\begin{aligned}\|\rho - g\|_{L^p(\mathcal{T})} &= \|\widetilde{\rho} \circ \mathcal{L} - \widetilde{g} \circ \mathcal{L}\|_{L^p(\mathcal{T})} \\ &\leq (\operatorname{diam}(\mathcal{T}))^{d/p} \|\widetilde{\rho} - \widetilde{g}\|_{L^p([0,1]^d)} \\ &\leq C_1 (\operatorname{diam}(\mathcal{T}))^{d/p} \omega_{\widetilde{g}}(r) \\ &\leq C_1 (\operatorname{diam}(\mathcal{T}))^{d/p} \omega_g^{\mathcal{T}}(\operatorname{diam}(\mathcal{T}) r).\end{aligned}$$

□

Lemma A.4. *Similarly, for a continuous mapping $\boldsymbol{f} = (f_1, \cdots, f_v)$ from a compact domain $\mathcal{T} \subset \mathbb{R}^d$ to $\mathbb{R}^v$, if there exist neural networks $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_v)$ that can approximate $\boldsymbol{f}$ then we have*

$$\|\boldsymbol{f} - \boldsymbol{\rho}\|_{L^p(\mathcal{T})} \leq v^{1/p} \cdot \max_{1 \leq i \leq v} \|f_i - \rho_i\|_{L^p(\mathcal{T})}$$

for $p \in [1,\infty]$.

Proof. It can be directly deduced from

$$\begin{aligned}\|\boldsymbol{f} - \boldsymbol{\rho}\|_{L^p(\mathcal{T})} &= \left(\int_E \sum_{i=1}^v |f_i(\mathbf{x}) - \rho_i(\mathbf{x})|^p \, \mathrm{d}\mathbf{x}\right)^{1/p} \\ \text{(Minkowski inequality)} &\leq \left(\sum_{i=1}^v \|f_i - \rho_i\|_{L^p(\mathcal{T})}^p\right)^{1/p} \\ &\leq \left(v \max_{1 \leq i \leq v} \|f_i - \rho_i\|_{L^p(\mathcal{T})}^p\right)^{1/p} \\ &= v^{1/p} \cdot \max_{1 \leq i \leq v} \|f_i - \rho_i\|_{L^p(\mathcal{T})}, \quad \text{for } p \in [1,\infty),\end{aligned}$$

and

$$\|\boldsymbol{f} - \boldsymbol{\rho}\|_{L^\infty(\mathcal{T})} = \max_{1 \leq i \leq v} \sup_{x \in \mathcal{T}} |f_i(\mathbf{x}) - \rho_i(\mathbf{x})| \leq \max_{1 \leq i \leq v} \|f_i - \rho_i\|_{L^\infty(\mathcal{T})}.$$

□

B. Proof of (i) of Theorem 1.1

Given the following theorem, we prove (i) of Theorem 1.1. The proof of Theorem B.1 is postponed to Sec. B.1.

Theorem B.1. *Given $f \in C\left([0,1]^d\right)$, for any $L \in \mathbb{N}^+$, there exists a function ρ implemented by a ReLU FNN with width $\max\{d+1, 5\}$ and depth $25L + 7d + 8$ such that $\|\rho\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and*

$$|f(\mathbf{x}) - \rho(\mathbf{x})| \leq 6\sqrt{d}\omega_f\left(L^{-2/d}\right), \quad \text{for any } \mathbf{x} \in [0,1]^d \backslash \Omega\left([0,1]^d, K, \delta\right),$$

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

Proof of (i) in Theorem 1.1. Assuming f is not a constant function since it is a trivial case, we have $\omega_f(r) > 0$ for any $r > 0$. We define $K = \lfloor L^{2/d} \rfloor$ and select a small $\delta \in \left(0, \frac{1}{3K}\right]$ such that

$$\leq Kd\delta\left(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\right)^p \leq \left(\omega_f\left(L^{-2/d}\right)\right)^p.$$

By Theorem B.1, there exists a ReLU FNN $\rho : \mathbb{R}^d \to \mathbb{R}$ with width $\max\{d+1, 5\}$ and depth $25L + 7d + 8$ such that $\|\rho\|_{L^\infty(\mathbb{R}^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and

$$|f(\mathbf{x}) - \rho(\mathbf{x})| \le 6\sqrt{d}\omega_f\left(L^{-2/d}\right), \quad \text{for any } \mathbf{x} \in [0,1]^d \backslash \Omega\left([0,1]^d, K, \delta\right),$$

Moreover, by the definition of the modulus of the continuity, we have $|f(\mathbf{x}) - f(\mathbf{0})| \le \omega_f(\sqrt{d})$ for any $\mathbf{x} \in [0,1]^d$, hence $\|f\|_{L^\infty([0,1]^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Then it follows from $\mu(\Omega([0,1]^d, K, \delta)$ that

$$\begin{aligned}
\|f - \rho\|^p_{L^p([0,1]^d)} &= \int_{\Omega([0,1]^d,K,\delta)} |f(\mathbf{x}) - \rho(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} + \int_{[0,1]^d \backslash \Omega([0,1]^d,K,\delta)} |f(\mathbf{x}) - \rho(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \\
&\le Kd\delta \left(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\right)^p + \left(6\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p \\
&\le \left(\omega_f\left(L^{-2/d}\right)\right)^p + \left(6\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p \le \left(7\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p.
\end{aligned}$$

Hence, $\|f - \rho\|_{L^p([0,1]^d)} \le 7\sqrt{d}\omega_f\left(L^{-2/d}\right)$.

□

B.1. Proof of Theorem B.1

Definition B.2. Let $K \in \mathbb{N}^+$, and δ be an arbitrary number in $\left(0, \frac{1}{3K}\right]$. For each d-dimensional index $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$, define $\mathbf{x}_{\boldsymbol{\beta}} := \boldsymbol{\beta}/K$ and

$$Q_{\boldsymbol{\beta}} := \left\{\mathbf{x} = (x_1, \cdots, x_d) \in [0,1]^d : x_i \in \left[\frac{\beta_i}{K}, \frac{\beta_i + 1}{K} - \delta \cdot 1_{\{\beta_i \le K-2\}}\right], i = 1, \cdots, d\right\}.$$

As is easy to see that $\mathbf{x}_{\boldsymbol{\beta}}$ is the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm and $[0,1]^d$ is divided into $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$ and $\Omega\left([0,1]^d, K, \delta\right)$, i.e.,

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

Now, given the following lemmas we show Theorem B.1. The proof of Lemma B.3 and B.3 can be found in Sec. B.2 and B.3 respectively.

Lemma B.3. *Let $K = \lfloor L^{2/d} \rfloor$, δ be an arbitrary number in $\left(0, \frac{1}{3K}\right]$ and $Q_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$ defined as it in Def. B.2. Then there exists a ReLU network $\Phi : \mathbb{R}^d \to \mathbb{R}$ width width $d+1$ and depth $4L + 7d - 4$ such that*

$$\Phi(\mathbf{x}) = \boldsymbol{\beta} \quad \textit{if } \mathbf{x} \in Q_{\boldsymbol{\beta}}$$

for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

Lemma B.4. *Let $K = \lfloor L^{2/d} \rfloor$, δ be an arbitrary number in $\left(0, \frac{1}{3K}\right]$ and $\mathbf{x}_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$ defined as it in Def. B.2. Assuming f is non-constant and $\widetilde{f} = f - f(\mathbf{0}) + \omega_f(\sqrt{d})$, then there exists a ReLU network $\phi \in \mathcal{NN}(5, 21L + 8)$ such that*

$$\left|\phi(\boldsymbol{\beta}) - \widetilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right)\right| \le \omega_f\left(\frac{\sqrt{d}}{K}\right), \tag{4}$$

for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$ and

$$0 \le \phi(\mathbf{x}) \le 2\omega_f(\sqrt{d}), \tag{5}$$

for any $\mathbf{x} \in \mathbb{R}^d$. Besides, $\phi = q \circ \psi$ where $\psi : \mathbb{R}^d \to \mathbb{R}$ is a linear function independent of f and $q : \mathbb{R} \to \mathbb{R}$ is a function depend on f.

Proof of Theorem B.1. Let $K = \lfloor L^{2/d} \rfloor$, and δ be an arbitrary number in $\left(0, \frac{1}{3K}\right]$. Now we divide $[0,1]^d$ into $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$ and $\Omega\left([0,1]^d, K, \delta\right)$ given by Definition B.2. We may assume f is not a constant since it is a trivial case and define $\widetilde{f} = f - f(\mathbf{0}) + \omega_f(\sqrt{d})$.

Let Φ and ϕ be the functions satisfying Lemma B.3 and Lemma B.4 respectively. We define the desired ReLU network ρ by $\rho := \phi \circ \Phi + f(\mathbf{0}) - \omega_f(\sqrt{d})$. Since $\Phi \in \mathcal{NN}(d+1, 4L+7d-4)$ and $\phi \in \mathcal{NN}(5, 21L+8)$, $\rho = \phi \circ \Phi + f(\mathbf{0}) - \omega_f(\sqrt{d})$ is in

$$\mathcal{NN}\left(\max\{d+1, 5\}, 25L + 7d + 4\right).$$

Now let us estimate the approximation error. Note that $f = \tilde{f} + f(\mathbf{0}) - \omega_f(\sqrt{d})$. By Equation (4), for any $\mathbf{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, we have

$$\begin{aligned}
|f(\mathbf{x}) - \rho(\mathbf{x})| &= \left|\widetilde{f}(\mathbf{x}) - \phi\left(\Phi(\mathbf{x})\right)\right| = \left|\widetilde{f}(\mathbf{x}) - \phi(\boldsymbol{\beta})\right| \\
&\leq \left|\widetilde{f}(\mathbf{x}) - \widetilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right)\right| + \left|\widetilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right) - \phi(\boldsymbol{\beta})\right| \\
&\leq \omega_f\left(\frac{\sqrt{d}}{K}\right) + \omega_f\left(\frac{\sqrt{d}}{K}\right) \leq 2\omega_f\left(2\sqrt{d}L^{-2/d}\right),
\end{aligned}$$

where the last inequality comes from the fact $K = \left\lfloor L^{2/d} \right\rfloor \geq \frac{L^{2/d}}{2}$ for any $L \in \mathbb{N}^+$. 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 $\mathbf{x} \in \cup_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d} Q_{\boldsymbol{\beta}} = [0,1]^d \backslash \Omega\left([0,1]^d, K, \delta\right)$, we have

$$|f(\mathbf{x}) - \rho(\mathbf{x})| \leq 2\omega_f\left(2\sqrt{d}L^{-2/d}\right) \leq 2\lceil 2\sqrt{d}\rceil \omega_f\left(L^{-2/d}\right) \leq 6\sqrt{d}\omega_f\left(L^{-2/d}\right).$$

It remains to show the upper bound of ρ. By Equation (5) and $\rho = \phi \circ \Phi + f(\mathbf{0}) - \omega_f(\sqrt{d})$, it holds that $\|\rho\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Thus, we finish the proof.

□

B.2. Proof of Lemma B.3

Given Proposition 3.3, we show Lemma B.3. The proof of Proposition 3.3 is postponed to Sec. B.5.

Proof of Lemma B.3. By Proposition 3.3, there exists a ReLU network $\hat{\zeta} = (x, \zeta(x)) \in \mathcal{NN}(2, 4L^{1/d} + 3)$ such that

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

It follows that $\zeta\left(x_i\right) = \beta_i$ if $\mathbf{x} = (x_1, x_2, \cdots, x_d) \in Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)$. Let

$$\begin{aligned}
&\mathcal{L}_0(x_1, \cdots, x_d) = (x_1, x_2, \cdots, x_d, 0), \\
&\varphi_i(x_1, x_2, \cdots, x_d, 0) = (x_1, x_2, \cdots, x_d, \zeta(x_i)), \quad \text{for } i = 1, 2, \cdots, d, \\
&\mathcal{L}_i(x_1, \cdots, x_d, y) = (x_1, \cdots, x_{i-1}, y, x_{i+1}, \cdots, x_d, 0), \quad \text{for } i = 1, 2, \cdots, d, \\
&\mathcal{L}(x_1, x_2, \cdots, x_d, 0) = (x_1, x_2, \cdots, x_d).
\end{aligned}$$

Then define $\Phi(x_1, \cdots, x_d) = \mathcal{L} \circ \mathcal{L}_d \circ \varphi_d \circ \cdots \circ \mathcal{L}_1 \circ \varphi_1 \circ \mathcal{L}_0(x_1, \cdots, x_d)$. It follows that Φ is a ReLU network in $\mathcal{NN}(d+1, 4dL^{1/d} + 3d) \subset \mathcal{NN}(d+1, 4L+7d-4)$ and

$$\Phi(\mathbf{x}) := \left(\zeta\left(x_1\right), \zeta\left(x_2\right), \cdots, \zeta\left(x_d\right)\right), \quad \text{for any } \mathbf{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$$

i.e., $\Phi(\mathbf{x}) = \boldsymbol{\beta}$ if $\mathbf{x} \in Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. Note that $\mathcal{NN}(d+1, 4dL^{1/d} + 3d) \subset \mathcal{NN}(d+1, 4L+7d-4)$ comes form the inequality $na^{1/n} \leq a + n - 1$ for any non-negative real number a and positive integer n. □

B.3. Proof of Lemma B.4

Given Proposition 3.4, we show Lemma B.4. The proof of Proposition 3.4 is postponed to Sec. B.6.

Proof of Lemma B.4. Let $K = \lfloor L^{2/d} \rfloor$, and δ be an arbitrary number in $\left(0, \frac{1}{3K}\right]$. Suppose $\mathbf{x}_{\boldsymbol{\beta}}$ is the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm and $[0,1]^d$ is divided into $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and $\Omega\left([0,1]^d, K, \delta\right)$ as Definition B.2.

We may assume f is not a constant since it is a trivial case. It is clear that $|f(\mathbf{x}) - f(\mathbf{0})| \leq \omega_f(\sqrt{d})$ for any $\mathbf{x} \in [0,1]^d$. For $\widetilde{f} = f - f(\mathbf{0}) + \omega_f(\sqrt{d})$, we have $0 \leq \widetilde{f}(\mathbf{x}) \leq 2\omega_f(\sqrt{d})$ for any $\mathbf{x} \in [0,1]^d$.

We want to construct ϕ mapping $\boldsymbol{\beta}$ approximately to $\tilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right)$. The construction of the sub-network ϕ_2 is essentially based on Proposition 3.4. To meet the requirements of applying Proposition 3.4, we follow a fact from (Shen et al., 2019a) as shown below.

Fact (Shen et al., 2019a): Let ψ be a linear function defined as

$$\psi(\mathbf{x}) := \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \mathbf{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d.$$

There exists a CPwL function $g : [0,1]^d \to \mathbb{R}$ that satisfies $g \circ \psi(\boldsymbol{\beta}) = \widetilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right)$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Besides, we have

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

and

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

Note $2K^d = 2\left(\lfloor L^{2/d} \rfloor\right)^d \leq 2\, L^2 \leq \widetilde{L}^2$, where $\widetilde{L} = \frac{3}{2}L$. Hence if we set $y_j = g\left(\frac{j}{2K^2}\right)$ and $\varepsilon = \omega_f\left(\frac{\sqrt{d}}{K}\right) > 0$ in Proposition 3.4, there exists a ReLU network $\tilde{\gamma} \in \mathcal{NN}(5, 14\widetilde{L} + 8)$ such that

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

and

$$0 \leq \tilde{\gamma}(x) \leq \max\left\{ g\left(\frac{j}{2K^d}\right) : j = 0, 1, \cdots, 2K^d - 1 \right\} \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}. \tag{6}$$

By defining $\gamma(x) := \tilde{\gamma}\left(2K^d x\right)$ for any $x \in \mathbb{R}$, we have $\gamma \in \mathcal{NN}(5, 21L + 8)$, $0 \leq \gamma(x) = \tilde{\gamma}\left(2K^d x\right) \leq 2\omega_f(\sqrt{d})$ for any $x \in \mathbb{R}$, and

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

Now define ϕ as $\phi := \gamma \circ \psi$ and note that $\psi : \mathbb{R}^d \to \mathbb{R}$ is a linear function and $\gamma \in \mathcal{NN}(5, 21L + 8)$. We have $\phi \in \mathcal{NN}(5, 21L + 8)$. Thus,

$$\left| \phi(\boldsymbol{\beta}) - \widetilde{f}\left(\mathbf{x}_{\boldsymbol{\beta}}\right) \right| = \left| \gamma\left(\psi(\boldsymbol{\beta})\right) - g\left(\psi(\boldsymbol{\beta})\right) \right| \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \tag{7}$$

for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Equation (6) and $\phi = \gamma \circ \psi$ implies

$$0 \leq \phi(\mathbf{x}) \leq 2\omega_f(\sqrt{d}), \tag{8}$$

for any $\mathbf{x} \in \mathbb{R}^d$.

□

B.4. Approximation of Mappings: Proof of (i) of Theorem 3.1

Now, we assume $\boldsymbol{f} = (f_1, f_2, \cdots, f_v)$ is a continuous mapping in from $[0,1]^d$ to $\mathbb{R}^v$. Similarly, we define $\widetilde{f}_i = f_i - f_i(\mathbf{0}) + \omega_{f_i}(\sqrt{d})$ for $i = 1, 2, \cdots, v$. From Lemma B.4, there exists a linear function $\psi : \mathbb{R}^d \to \mathbb{R}$ and $\gamma^{(i)} : \mathbb{R} \to \mathbb{R}$ such that $\phi_i = \gamma^{(i)} \circ \psi$ and

$$\left|\phi_i(\boldsymbol{\beta}) - \widetilde{f}_i\left(\mathbf{x}_{\boldsymbol{\beta}}\right)\right| \leq \omega_{f_i}\left(\frac{\sqrt{d}}{K}\right), \tag{9}$$

for any $\beta \in \{, 1, \cdots, K-1\}^d$ where $\gamma^{(i)} : \mathbb{R} \to \mathbb{R}$ is a ReLU network in $\mathcal{NN}(5, 21L+8)$. Hence ϕ_i is a ReLU network in $\mathcal{NN}^{1,1}(5, 21L+8)$. Moreover, for $x \in \mathbb{R}^d$

$$0 \leq \phi_i(\mathbf{x}) \leq 2\omega_f(\sqrt{d}) \tag{10}$$

for $i = 1, 2, \cdots, v$.

Let $\rho_i := \phi_i \circ \Phi + f_i(\mathbf{0}) - \omega_f(\sqrt{d})$. It follows from the proof of Theorem B.1 that for any $x \in [0,1]^d \backslash \Omega\left([0,1]^d, K, \delta\right)$

$$|f_i(\mathbf{x}) - \rho_i(\mathbf{x})| \leq 6\sqrt{d}\omega_{f_i}\left(L^{-2/d}\right) \leq 6\sqrt{d}\omega_f\left(L^{-2/d}\right) \tag{11}$$

for $i = 1, 2, \cdots, v$. Moreover, we have

$$\|\rho_i\|_{L^\infty(\mathbb{R}^d)} \leq |f_i(\mathbf{0})| + \omega_f(\sqrt{d})$$

for $i = 1, 2, \cdots, v$.

Let $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_v)$. Next, we explain how to compute $\boldsymbol{\rho}$ by a ReLU neural network. Assume we have computed $\Phi(\mathbf{x})$ by Lemma B.3. Then we can use one more channel to reserve the value of $\psi(\Phi(\mathbf{x}))$. We can then use 6 channels to compute ϕ_i one by one and store the value of $\rho_i (i = 1, 2, \cdots, v)$ in v channels. Finally, we can output $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_v)$. For each ϕ_i, its depth is $21L + 8$ by Lemma B.4 and recall that $\Phi \in \mathcal{NN}(d+1, 4L+7d-4)$. Thus, $\boldsymbol{\rho} \in \mathcal{NN}^{d,v}\left(\max\{d+1, v+6\}, (4+21v)L + 7d + 8v - 4\right)$.

Last, we end this section by extending the domain to $[0,1]^d$. With the similar idea as the proof B, we set $K = \left\lfloor L^{2/d} \right\rfloor$ and choose a small $\delta \in \left(0, \frac{1}{3K}\right]$ such that

$$Kd\delta\left(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\right)^p \leq \left(\omega_f\left(L^{-2/d}\right)\right)^p.$$

By the above discussion, for $i = 1, 2, \cdots, v$,

$$\|\rho_i\|_{L^\infty(\mathbb{R}^d)} \leq |f_i(\mathbf{0})| + \omega_f(\sqrt{d})$$

and

$$|f_i(\mathbf{x}) - \rho_i(\mathbf{x})| \leq 6\sqrt{d}\omega_f\left(L^{-2/d}\right), \quad \text{for any } \mathbf{x} \in [0,1]^d \backslash \Omega\left([0,1]^d, K, \delta\right),$$

It follows from $\mu\left(\Omega\left([0,1]^d, K, \delta\right)\right) \leq Kd\delta$, $\|\rho_i\|_{L^\infty([0,1]^d)} \leq |f_i(\mathbf{0})| + \omega_f(\sqrt{d})$, and the proof of Theorem 1.1 that

$$\begin{aligned}
\|f_i - \rho_i\|_{L^p([0,1]^d)} &= \int_{\Omega([0,1]^d, K, \delta)} |f_i(\mathbf{x}) - \rho_i(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} + \int_{[0,1]^d \backslash \Omega([0,1]^d, K, \delta)} |f_i(\mathbf{x}) - \rho_i(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \\
&\leq Kd\delta\left(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\right)^p + \left(6\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p \\
&\leq \left(\omega_f\left(L^{-2/d}\right)\right)^p + \left(6\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p \\
&\leq \left(7\sqrt{d}\omega_f\left(L^{-2/d}\right)\right)^p.
\end{aligned}$$

Hence, $\|f_i - \rho_i\|_{L^p([0,1]^d)} \leq 7\sqrt{d}\omega_f\left(L^{-2/d}\right)$ for $i = 1, 2, \cdots, v$. By Lemma A.3 and A.4, we have finished the proof of (i) of Theorem 3.1.

B.5. Proof of Proposition 3.3

We first give some lemmas so that it can be convenient for us to state our proof.

Definition B.5. A function $g : \mathbb{R}^d \to \mathbb{R}^v$ is a max-min string of length $L \geq 1$ on d input variables and v output variables if there exist affine functions $\ell_1, \ldots, \ell_L : \mathbb{R}^d \to \mathbb{R}^v$ such that

$$g = \tau_{L-1}\left(\ell_L, \tau_{L-2}\left(\ell_{L-1}, \ldots, \tau_2\left(\ell_3, \tau_1\left(\ell_1, \ell_2\right)\right)\cdots\right)\right),$$

where each τ_i is either a coordinate-wise max or a min.

Proposition B.6 (Proposition 2, (Hanin & Sellke, 2017))**.** *For any input $\mathbf{x} \in \mathbb{R}^d$ and any max-min string $g : \mathbb{R}^d \to \mathbb{R}^v$ of length L, there is a network $\tilde{g} \in \mathcal{I}_{\mathrm{ReLU}}^{d,v}(d, v, 0; L)$ that can generate g, i.e., $\tilde{g} = g$. This implies that there is a ReLU network $\phi \in \mathcal{NN}^{d,d+v}\left(d + v, L\right)$ such that $\phi(x) = (x, g(x))$.*

Lemma B.7. *Let f any given univariate continuous piecewise linear function with L breakpoints: $x_1 < x_2 < \cdots < x_L$. For $i = 0, 1, \cdots, L$, we define $f_i(x)$ to be the linear function coincides with the linear function $f(x)$ in $[x_i, x_{i+1}]$ and is extended to $\mathbb{R}$. Here we set $x_0 = -\infty$ and $x_{L+1} = \infty$. Assume f satisfies the following property for $i = 1, 2, \cdots, L$:*

(i) if $f_i'(x) \geq f_{i-1}'(x)$, then $f_i(x) \leq f(x), \forall x \leq x_i$, and

(ii) if $f_i'(x) < f_{i-1}'(x)$, then $f_i(x) \geq f(x), \forall x \leq x_i$.

Then there exists a ReLU network ϕ with width 2 and depth L such that $\phi(x) = (x, f(x))$.

Proof. It is easy to show f is a max-min string of length L by induction. Let us briefly show it. $L = 1$ is the trivial case. If f is CPwL of two breakpoints and satisfies the assumption (actually, any CPwL function of two breakpoints will satisfy the condition), then we have $f(x) = \max\{f_0(x), f_1(x)\}$ for the case (i) and $f(x) = \min\{f_0(x), f_1(x)\}$ for the case (ii). We now assume Lemma B.7 holds for $L = k$ we will show it also holds for $L = k + 1$. Suppose f is a CPwL function with $k + 1$ breakpoints ($x_1 < x_2 < \cdots < x_{k+1}$), we define g is the function such that $g(x)$ equal to $f(x)$ as $x < x_{k+1}$ and $g(x)$ is equal to $f_k(x)$ as $x > x_{k+1}$. Then $g(x)$ is a CPwL function satisfying the assumption, hence g is a max-string of length k. Moreover, for $i = k + 1$, we have $f(x) = \max\{g(x), f_{k+1}(x)\}$ if (i) is the case and $f(x) = \min\{g(x), f_{k+1}(x)\}$ if (ii) is the case. Thus, f is a max-min string of length L by induction. Then we can get the conclusion by Proposition B.6. □

Now, we are ready to prove Proposition 3.3 and we may prove Proposition 3.3 with the following notation.

Proposition B.8 (Proposition 3.3)**.** *For any $L, d \in \mathbb{N}^+$ and $\delta \in \left(0, \frac{1}{3K}\right]$ with $K = \left\lfloor L^{2/d} \right\rfloor$, there exists a function $\hat{\rho} : [0, 1] \to \mathbb{R}^2, \hat{\rho}(x) = (x, \rho(x))$ implemented by a ReLU network with width 2 and depth not more than $4L^{\frac{1}{d}} + 3$ such that*

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

proof of Prop. 3.3. Without loss of generality, assume $K = \left\lfloor L^{2/d} \right\rfloor = \widetilde{L}^2$ where $\widetilde{L} = L^{1/d}$. We first consider the sample set

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

By Lemma B.7, there exists a ReLU network $\hat{\phi}_1 \in \mathcal{NN}(2, 2\widetilde{L} + 1)$ such that $\hat{\phi}_1(x) = (x, \phi_1(x))$ where $\phi_1(x)$ is a CPwL function with breakpoints in the above point set, i.e.,

- $\phi_1\left(\frac{\widetilde{L}-1}{\widetilde{L}}\right) = \phi_1(1) = \widetilde{L} - 1$ and $\phi_1\left(\frac{m}{\widetilde{L}}\right) = \phi_1\left(\frac{m+1}{\widetilde{L}} - \delta\right) = m$ for $m = 0, 1, \cdots, \widetilde{L} - 2$, and
- ϕ_1 is linear on $\left[\frac{\widetilde{L}-1}{\widetilde{L}}, 1\right]$ and each interval $\left[\frac{m}{\widetilde{L}}, \frac{m+1}{\widetilde{L}} - \delta\right]$ for $m = 0, 1, \cdots, \widetilde{L} - 2$.

Then we have

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

Next we consider the another sample set

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

Its size is $2\widetilde{L}+1$. By Lemma B.7, there exists a ReLU network $\phi_2 \in \mathcal{NN}(2, 2\widetilde{L}+1)$ such that

- $\phi_2\left(\frac{\widetilde{L}-1}{\widetilde{L}^2}\right) = \phi_2\left(\frac{1}{\widetilde{L}}\right) = \widetilde{L}-1$ and $\phi_2\left(\frac{\ell}{\widetilde{L}^2}\right) = \phi_2\left(\frac{\ell+1}{\widetilde{L}^2} - \delta\right) = \ell$ for $\ell = 0, 1, \cdots, \widetilde{L}-2$;
- ϕ_2 is linear on $\left[\frac{\widetilde{L}-1}{\widetilde{L}^2}, \frac{1}{\widetilde{L}}\right]$ and each interval $\left[\frac{\ell}{\widetilde{L}^2}, \frac{\ell+1}{\widetilde{L}^2} - \delta\right]$ for $\ell = 0, 1, \cdots, \widetilde{L}-2$.

It follows that, for $m = 0, 1, \cdots, \widetilde{L}-1$ and $\ell = 0, 1, \cdots, \widetilde{L}-1$,

$$\phi_2\left(x - \frac{m}{\widetilde{L}}\right) = \ell, \quad \text{for } x \in \left[\frac{m\widetilde{L}+\ell}{\widetilde{L}^2}, \frac{m\widetilde{L}+\ell+1}{\widetilde{L}^2} - \delta \cdot \mathbb{1}_{\{\ell \le \widetilde{L}-2\}}\right].$$

The fact $K = \widetilde{L}^2$ implies each $k \in \{0, 1, \cdots, K-1\}$ can be unique represented by $k = m\widetilde{L} + \ell$ for $m = 0, 1, \cdots, \widetilde{L}-1$ and $\ell = 0, 1, \cdots, \widetilde{L}-1$. For any $x \in \left[\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\right]$ for $k \in \{0, 1, \cdots, K-1\}$, $\hat{\phi}_1(x) = (x, \phi(x)) = (x, m)$. Next, we define an affine mapping ϕ_0 such that $\phi_0(x, m) = \left(x - \frac{m}{\widetilde{L}}, m\right)$. Finally, let $\hat{\phi}_2(x) = (\phi_2(x), x)$. Then $\hat{\phi}_2\left(x - \frac{m}{\widetilde{L}}, m\right) = \left(\phi_2\left(x - \frac{m}{\widetilde{L}}\right), m\right) = (\ell, m)$. With a final affine layer $\mathcal{L}(\ell, m) = m\widetilde{L} + \ell = k$.

Thus, the desired function $\rho := \mathcal{L} \circ \hat{\phi}_2 \circ \phi_0 \circ \hat{\phi}_1$ can be implemented by a ReLU FNN and

$$\rho(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\right] \text{ for } k \in \{0, 1, \cdots, K-1\}.$$

Moreover, $\rho \in \mathcal{NN}(2, 4\widetilde{L}+3)$ so we finish the proof.

□

B.6. Proof of Propsition 3.4

Lemma B.9 (Proposition 4, (Hanin & Sellke, 2017))**.** *Let $S \subseteq \mathbb{R}$ be a finite set. Then any function $f : S \to \mathbb{R}$ can be computed exactly by a max-min string of length $2|S|$. This implies there exists a ReLU network ϕ with width 2 and depth $2|S|$ such that $\phi(x) = (x, f(x))$.*

Lemma B.10. *For any $L \in \mathbb{N}^+$ and $q \in \{2, 3\}$, there exists a ReLU network ϕ in $\mathcal{NN}(5, (2q+2)L)$ such that, for any $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1, \cdots, q-1\}$, we have*

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

Proof. Given $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1, \cdots, q-1\}$, define

$$\xi_j := 0.\theta_j\theta_{j+1}\cdots\theta_L = \sum_{i=j}^{L} \frac{\theta_i}{q^{i-j+1}}, \quad \text{for } j = 1, 2, \cdots, L$$

and $\mathcal{S}_q(x) : [0, q) \to \mathbb{R}$ as

$$\mathcal{S}_q(x) = k, \quad \text{if } x \in [k, k+1-\epsilon] \text{ for } k = 0, 1, \cdots, q-1,$$

where ϵ is a parameter to be determined later. Then we have

$$\theta_j = \lfloor q\xi_j \rfloor \text{ for } j = 1, 2, \cdots, L,$$

and

$$\xi_{j+1} = q\xi_j - \theta_j \text{ for } j = 1, 2, \cdots, L-1.$$

Moreover, $\lfloor \cdot \rfloor$ can be approximated by the ReLU network $\mathcal{S}_q(\cdot)$. Note that if $\epsilon < q^{-L}$, then $\lfloor qx \rfloor = \mathcal{S}_q(qx)$ for any $x = 0.\theta_1\theta_2 \cdots \theta_l$ where $l \leq L$. Now let

$$\mathcal{S}(x) := \begin{cases} 1, x \geq 0, \\ 0, x < 0. \end{cases}$$

If $x \in \mathbb{Z}$, $\mathcal{S}(x) = \mathcal{S}_0(x)$ where $\mathcal{S}_0(x)$ is defined by

$$\mathcal{S}_0(x) := \begin{cases} 1, x \geq 0, \\ x+1, 0 \geq x \geq -1, \\ -1, x < -1. \end{cases}$$

Now we have

$$\sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^{L} \theta_j \mathcal{S}(\ell - j) = \sum_{j=1}^{L} \theta_j \mathcal{S}_0(\ell - j) := \sum_{j=1}^{L} z_{\ell,j}$$

for $\ell = 1, 2, \cdots, L$ where $z_{\ell,j} = \theta_j \mathcal{S}_0(\ell - j) \geq 0$.

Here the multiplication of $x \in \{0, 1\}$ and $y \in \{0, 1\}$ can be done by $xy = \sigma(x + y - 1)$, and the multiplication of $x \in \{0, 1, 2\}$ and $y \in \{0, 1\}$ can be done by $xy = \sigma(x + y - 1) - \sigma(x - y - 1)$.

Now, we construct a ReLU network ϕ such that $\phi(0.\theta_1\theta_2 \cdots \theta_L, \ell) = \sum_{j=1}^{\ell} \theta_j$ for $\ell = 1, 2, \cdots, L$. Note that $\mathcal{S}_q$ is a linear interpolation at the sample set

$$\{(k, k) : k = 0, 1, \cdots, q-1\} \cup \{(k+1-\epsilon, k) : k = 0, 1, \cdots, q-1\}.$$

Then by Lemma B.7, $\mathcal{S}_q(q \geq 1)$ can be implemented by a ReLU network $\mathcal{L}$ with width 2 and depth $2q$, i.e., $\mathcal{L}(x) = (x, \mathcal{S}_q(x))$. Similarly, $\mathcal{S}_0$ can be implemented by a ReLU network with width 2 and depth 2. Let $\mathcal{A}^{(j)}(\xi_j) = (\xi_j, \theta_j = \mathcal{S}_q(q\xi_j))$ and $\mathcal{B}^{(j)}(\ell) = (\ell, \mathcal{S}_0(\ell - j))$ for $j = 1, 2, \cdots, L$. Here note that $\mathcal{A}(ax + b)$ is a CPwL function of x if $\mathcal{A}(x)$ is CPwL. Then we have the following network architecture to output the desired value:

$$\begin{pmatrix} \xi_1 \\ \ell \end{pmatrix} \to \begin{pmatrix} \xi_1 \\ \ell \\ \theta_1 \\ \mathcal{S}_0(\ell - 1) := y_1 \\ 0 \end{pmatrix} \to \begin{pmatrix} \xi_2 \\ \ell \\ \sigma(\theta_1 + y_1 - 1) \\ \sigma(\theta_1 - y_1 - 1) \\ 0 \end{pmatrix} \to \begin{pmatrix} \xi_2 \\ \ell \\ \sigma(\theta_1 + y_1 - 1) \\ \sigma(\theta_1 - y_1 - 1) \\ z_{\ell,1} \end{pmatrix} \to \cdots \to \begin{pmatrix} \xi_m \\ \ell \\ - \\ - \\ \sum_{j=1}^{m-1} z_{\ell,j} \end{pmatrix}$$

$$\to \begin{pmatrix} \xi_m \\ \ell \\ \theta_m \\ \mathcal{S}_0(\ell - m) : y_m \\ \sum_{j=1}^{m-1} z_{\ell,j} \end{pmatrix} \to \begin{pmatrix} \xi_{m+1} \\ \ell \\ \sigma(\theta_m + y_m - 1) \\ \sigma(\theta_m - y_m - 1) \\ \sum_{j=1}^{m-1} z_{\ell,j} \end{pmatrix} \to \begin{pmatrix} \xi_{m+1} \\ \ell \\ \sigma(\theta_m + y_m - 1) \\ \sigma(\theta_m - y_m - 1) \\ \sum_{j=1}^{m} z_{\ell,j} \end{pmatrix} \to \cdots \to \begin{pmatrix} - \\ \ell \\ - \\ - \\ \sum_{j=1}^{L} z_{\ell,j} \end{pmatrix} \to \sum_{j=1}^{\ell} \theta_j$$

The entire network has not more than $(2q + 2)L$ layers and its width is 5.

□

Lemma B.11. *For any $L \in \mathbb{N}^+, q \in \{2,3\}$, any $\theta_{k,\ell} \in \{0,1,\cdots,q-1\}$ for $k,\ell = 0,1,\cdots,L-1$, there exists a ReLU FNN $\phi : \mathbb{R}^2 \to \mathbb{R}$ with width 5 and depth $(2q+4)L+2$ such that*

$$\phi(k,\ell) = \sum_{j=0}^{\ell} \theta_{k,j}, \quad \text{for } k,\ell = 0,1,\cdots,L-1$$

Proof. Let

$$y_k := 0.\theta_{k,0}\theta_{k,1}\cdots\theta_{k,L-1}, \quad \text{for } k = 0,1,\cdots,L-1.$$

Next, we consider the sample set $\{(k,y_k) : k = 0,1,\cdots,L\}$, whose size is $L+1$. By Lemma B.9, there exists a ReLU network $\phi_1 \in \mathcal{NN}(2,2L+2)$ such that

$$\phi_1(k) = y_k, \quad \text{for } k = 0,1,\cdots,L-1.$$

By Lemma B.10, there exists $\phi_2 \in \mathcal{NN}(5,(2q+2)L)$ such that, for any $\xi_1,\xi_2,\cdots,\xi_L \in \{0,1,\cdots,q-1\}$, we have

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

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

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

Thus, for $k,\ell = 0,1,\cdots,L-1$, we have

$$\phi_2\left(\phi_1(k),\ell+1\right) = \phi_2\left(y_k,\ell+1\right) = \phi_2\left(0.\theta_{k,0}\theta_{k,1}\cdots\theta_{k,L-1},\ell+1\right) = \sum_{j=0}^{\ell}\theta_{k,j}$$

We have ϕ is a ReLU network of width 5 and depth $(2q+2)L+2L+2$. □

Lemma B.12. *For any $\varepsilon > 0, L \in \mathbb{N}^+$, assume $\{y_{k,\ell} \geq 0 : k,\ell = 0,1,\cdots,L-1\}$ is a sample set with*

$$|y_{k,\ell} - y_{k,\ell-1}| \leq \varepsilon, \quad \text{for } k,\ell = 0,1,\cdots,L-1.$$

Then there exists a ReLU netework $\phi \in \mathcal{NN}(5,12L+6)$ such that

(i) $|\phi(k,\ell) - y_{k,\ell}| \leq \varepsilon$, *for $k,\ell = 0,1,\cdots,L-1$, and*

(ii) $0 \leq \phi\left(x_1,x_2\right) \leq \max\left\{y_{k,\ell} : k,\ell = 0,1,\cdots,L-1\right\}$, *for any $x_1,x_2 \in \mathbb{R}$.*

Proof. Define

$$a_{k,\ell} := \lfloor y_{k,\ell}/\varepsilon \rfloor, \quad \text{for } k,\ell = 0,1,\cdots,L-1.$$

We will construct a a ReLU FNN to map the index (k,ℓ) to $a_{k,\ell}\varepsilon$ for $k,\ell = 0,1,\cdots,L-1$.

Define $b_{k,0} := 0$ and $b_{k,\ell} := a_{k,\ell} - a_{k,\ell-1}$ for $k,\ell = 0,1,\cdots,L-1$. Since $|y_{k,\ell} - y_{k,\ell-1}| \leq \varepsilon$ for all k and ℓ, we have $b_{k,\ell} \in \{-1,0,1\}$. Hence, we have

$$a_{k,\ell} = a_{k,0} + \sum_{j=1}^{\ell}\left(a_{k,j} - a_{k,j-1}\right) = a_{k,0} + \sum_{j=1}^{\ell} b_{k,j} = a_{k,0} + \sum_{j=0}^{\ell} b_{k,j}$$

for $k = 0, 1, \cdots, L-1$ and $\ell = 1, \cdots, L-1$. For the sample set $\{(k, a_{k,0}) : k = 0, 1, \cdots, L-1\} \cup \{(L, 0)\}$, whose size is $L+1$, by Lemma B.9, there exists a ReLU network $\tilde{\psi}_1 \in \mathcal{NN}(2, 2L+2)$ such that

$$\tilde{\psi}_1(k) = (k, \psi_1(k) = a_{k,0}), \quad \text{for } k = 0, 1, \cdots, L-1.$$

It follows that there exists a ReLU network $\hat{\psi}_1 \in \mathcal{NN}(3, 2L+1)$ such that $\hat{\psi}_1(k, \ell) = (k, \ell, a_{k,0})$.

By Lemma B.11, there exists a ReLU network $\psi_2 \in \mathcal{NN}(5, 10L+2)$ such that

$$\psi_2(k, \ell) = \sum_{j=0}^{\ell} b_{k,j} = \sum_{j=0}^{\ell} (b_{k,j} + 1) - \ell.$$

Here note that $b_{k,j} \in \{0, 1, 2\}$ which will satisfy the condition of Lemma B.11.

Thus, we can compute $a_{k,0}$ first by $\hat{\psi}_1$. According to the construction in Lemma B.10 and Lemma B.11 we use one channel to reserve the value of $a_{k,0}$ and the partial sum $\sum_{j=0}^{\ell} b_{k,j}$ computed by ψ_2. Thus, there exists a ReLU FNN $\hat{\psi}_2$ with width 5 and depth $10L+2$ such that

$$\hat{\psi}_2(k, \ell, a_{k,0}) = a_{k,0} + \psi_2(k, \ell) = a_{k,0} + \sum_{j=0}^{\ell} b_{k,j} = a_{k,\ell}.$$

Define

$$\phi_1(k, \ell) = \mathcal{L} \circ \hat{\psi}_2 \circ \hat{\psi}_1(k, \ell) = \mathcal{L} \circ \hat{\psi}_2(k, \ell, a_{k,0}) = \varepsilon a_{k,\ell}$$

by choosing appropriate affine mapping $\mathcal{L}$. Then ϕ_1 is a ReLU network with width 5 and depth $12L+4$ and $\phi_1(k, \ell) = \varepsilon a_{k,l}$ for $k = 0, 1, \cdots .L-1$ and $\ell = 0, 1, \cdots, L-1$. Let $\phi_2(x) = \min\{\sigma(x), y_{\max}\}$. Then for $x, y \in \mathbb{R}$, $\phi(x, y) := \phi_2 \circ \phi_1(x, y) \le y_{\max}$ and $\phi(k, \ell) = \min\{\varepsilon a_{k,\ell}, y_{\max}\} = a_{k,\ell}$ for $k, \ell = 0, 1, \cdots, L$.

Note $\min\{a, b\} = a - (a-b)_+$. Then ϕ is a ReLU network in $\mathcal{NN}(5, 12L+6)$.

□

Now, we are ready to prove Prop. 3.4. We are going to prove the Proposition 3.4 with the following notation.

Proposition B.13 (Proposition 3.4)**.** *Given any $\varepsilon > 0$ and arbitrary $L, J \in \mathbb{N}^+$ with $J \le L^2$, assume $y_j \ge 0$ for $j = 0, 1, \cdots, J-1$ are samples with*

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

Then there exists a ReLU network $\rho \in \mathcal{NN}(5, 14L+8)$ such that

(i) $|\rho(j) - y_j| \le \varepsilon$ *for* $j = 0, 1, \cdots, J-1$*, and*

(ii) $0 \le \rho(x) \le \max\{y_j : j = 0, 1, \cdots, J-1\}$ *for any* $x \in \mathbb{R}$.

Proof of Prop. 3.4. Without loss of generality, assume $J = L^2$ since we can set $y_{J-1} = y_J = y_{J+1} = \cdots = y_{L^2-1}$ if $J < L^2$. For the sample set

$$\{(kL, k) : k = 0, 1, \cdots, L\} \cup \{(kL + L - 1, k) : k = 0, 1, \cdots, L-1\}$$

whose size is $2L+1$. We then have a ReLU network $\hat{\phi}_1 \in \mathcal{NN}(2, 2L+1)$ by Lemma B.7 such that

- $\hat{\phi_1}(x) = (\phi_1(x), x)$,
- $\phi_1(L^2) = L$ and $\phi_1(kL) = \phi_1(kL + L - 1) = k$ for $k = 0, 1, \cdots, L-1$, and
- ϕ_1 is a CPwL function with breakpoints that coincide exactly with the first coordinate of the elements in the sample set.

It follows that

$$\phi_1(j) = k, \quad \text{and} \quad j - L\phi_1(j) = \ell, \quad \text{where } j = kL + \ell,$$

for $k, \ell \in \{0, 1, \cdots, L-1\}$. Note that any number j in $\{0, 1, \ldots, J-1\}$ can be uniquely indexed as $j = kL + \ell$ for $k = 0, 1, \cdots, L-1$ and $\ell = 0, 1, \cdots, L-1$. So we can denote $y_j = y_{kL+\ell}$ as $y_{k,\ell}$. Then by Lemma B.12, there exists $\phi_2 \in \mathcal{NN}(5, 12L+6)$ such that

$$|\phi_2(k, \ell) - y_{k,\ell}| \leq \varepsilon, \quad \text{for } k, \ell = 0, 1, \cdots, L-1,$$

and

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

So any $j = kL + \ell \in \{0, 1, \cdots, J-1\}$, we can have an affine mapping $\mathcal{L}$ such that

$$\mathcal{L} \circ \hat{\phi}_1(j) = \mathcal{L}(k, j) = (k, \ell).$$

Let $\rho = \phi_2 \circ \mathcal{L} \circ \phi_1$. We have ρ is a ReLU network in $\mathcal{NN}(5, 14L+8)$ and it satisfy

$$|\rho(j) - y_j| \leq \varepsilon.$$

Then we have finished the proof.

□

C. Proof of (ii) of Theorem 1.1

We follow the proof of theorem 2 in (Yarotsky, 2018). In our proof, we skip some details and focus on the constructions of narrow networks. All the overlooked details can be found in (Yarotsky, 2018). First, we recall the idea of achieving the optimal approximation rate (Yarotsky, 2018).

C.1. Preliminaries: Key Steps in (Yarotsky, 2018)

Step 1: Space Partitions.

Generally, we will approximate f by interpolation of f on a scale $1/N$. To this end, divide $[0,1]^d$ into standard simplexes on the grid $(\frac{\mathbb{Z}}{N})^d$. Each simplex is a triangle

$$\Delta_{\mathbf{n},\pi}^{(N)} = \left\{ \mathbf{x} \in \mathbb{R}^d : 0 \leq x_{\pi(1)} - \frac{n_{\pi(1)}}{N} \leq \ldots \leq x_{\pi(d)} - \frac{n_{\pi(d)}}{N} \leq \frac{1}{N} \right\}$$

where $\mathbf{n} = (n_1, \ldots, n_d) \in \mathbb{Z}^d$ and π is a permutation of d elements. Denote by P_N the set of all simplexes on the grid $(\mathbb{Z}/N)^d$. The vertices of these simplexes are the points of the grid $(\mathbb{Z}/N)^d$. The set of all the vertices is called the N-grid and a particular vertex is called an N-knot. For an N-knot we call the union of simplexes it belongs to an N-patch. There is an illustration figure for these notations in (Yarotsky & Zhevnerchuk, 2020).

Step 2: Function approximation.

Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be the "spike" function defined as the continuous piecewise linear function such that:
1. ϕ is linear on every simplex from the triangulation $\mathcal{P}_1$;
2. $\phi(\mathbf{0}) = 1, \phi(\mathbf{n}) = 0$ for all other $\mathbf{n} \in \mathbb{Z}^d$.

Then from formula (6) in (Yarotsky, 2018)

$$\phi(\mathbf{x}) = \left(\min \left(\min_{k \neq s} (1 + x_k - x_s), \min_k (1 + x_k), \min_k (1 - x_k) \right) \right)_+ . \tag{12}$$

Now, the piecewise linear interpolation $\tilde{f}_1$ on a scale $1/N$ is defined as

$$\tilde{f}_1(\mathbf{x}) = \sum_{\mathbf{n} \in \{0,1,\ldots,N\}^d} f\left(\frac{\mathbf{n}}{N}\right) \phi(N\mathbf{x} - \mathbf{n}) \tag{13}$$

with the approximation error

$$\|f-\tilde{f}_1\|_{L^\infty} \leq 3d\omega_f(\frac{1}{N}).$$

If we simply store each coefficient $f(\mathbf{n}/N)$ by $\mathcal{O}(1)$ parameters and construct a neural network with $W = \mathcal{O}(N^d)$ parameters to generate $\tilde{f}_1$, we can only achieve the sub-optimal approximation rate. To achieve the optimal approximation rate, we need to consider the interpolation function $p(\mathbf{x})$ on a more refined scale $1/M$ where $M < N$ and use networks with the same order parameters $W = \mathcal{O}(N^d)$ to generate this interpolation function $p(\mathbf{x})$.

To this end, we consider the approximation of the discrepancy $f_2 = f - \tilde{f}_1$. Further, f_2 can be represented by a finite sum of functions with supports consisting of disjoint N-patches. Concretely,

$$f_2 = \sum_{\mathbf{q}\in\{0,1,2\}^d} f_{2,\mathbf{q}}, \tag{14}$$

where

$$f_{2,\mathbf{q}} = f_2 g_{\mathbf{q}} \quad \text{and} \quad g_{\mathbf{q}}(\mathbf{x}) = \sum_{\mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^d)\cap[0,N]^d} \phi(N\mathbf{x}-\mathbf{n}). \tag{15}$$

Moreover, $g_{\mathbf{q}}$ satisfy $1 = \sum_{\mathbf{q}\in\{0,1,2\}^d} g_{\mathbf{q}}$. Each function $f_{2,\mathbf{q}}$ is supported on the disjoint union of cubes $Q_{\mathbf{n}} = X_{s=1}^d \left[\frac{n_s-1}{N}, \frac{n_s+1}{N}\right]$ with $\mathbf{n} \in \left(\mathbf{q}+(3\mathbb{Z})^d\right) \cap [0,N]^d$ corresponding to the spikes in the expansion (15).

Recall that our budget is $W = \mathcal{O}(N^d)$ parameters and layers. We then write $N = \left\lfloor c_1 W^{1/d} \right\rfloor$ and we further set $M = c_2 W^{2/d}$. Without loss of generality, we assume M/N is an integer. We then consider a more refined approximation on scale $1/M$. We define $\tilde{f}_{2,\mathbf{q}}$ to be piecewise linear with respect to the refined triangulation P_M and to be given on the refined grid $(\mathbb{Z}/M)^d$ by

$$\tilde{f}_{2,\mathbf{q}}\left(\frac{\mathbf{m}}{M}\right) = \lambda \left\lfloor f_{2,\mathbf{q}}\left(\frac{\mathbf{m}}{M}\right)/\lambda \right\rfloor, \quad \mathbf{m} \in [0,\ldots,M]^d. \tag{16}$$

Here the parameter λ is given by

$$\lambda = \left(6d^{3/2}+1\right)\omega_f\left(\frac{1}{M}\right). \tag{17}$$

Then the full approximation of f is

$$\rho = \tilde{f}_1 + \tilde{f}_2 = \tilde{f}_1 + \sum_{\mathbf{q}\in\{0,1,2\}^d} \tilde{f}_{2,\mathbf{q}} \tag{18}$$

Step 3: Accuracy of the Full Approximation.

According to equation (10) in (Yarotsky, 2018), we have

$$\|f-\rho\|_{L^\infty} \leq 3^d(3d+1)\left(6d^{3/2}+1\right)\omega_f\left(\frac{1}{M}\right). \tag{19}$$

Then if we can generate ρ by a ReLU network of $\mathcal{O}(W)$ parameters, we can achieve the optimal approximation rate as follows:

$$\|f-\rho\|_{L^\infty} = \mathcal{O}\left(\omega_f\left(\mathcal{O}\left(W^{-2/d}\right)\right)\right). \tag{20}$$

In (Yarotsky, 2018), the author constructs a network of width $2d+10$ to achieve this rate. However, the width $2d+10$ is slightly larger than the state-of-the-art minimum width to satisfy the universality, which is $d+1$. Thus, in this paper, we aim to construct a network of width $d+\mathcal{O}(1)$ to achieve this rate. Before proceeding to this proof, we need to reformulate ρ so that it can be convenient for us to construct a ReLU network to generate it. It suffices to reformulate $\tilde{f}_{2,\mathbf{q}}$ since $\tilde{f}_1$ is easy to be generated by a ReLU network.

Step 4: Reformulation of $\tilde{f}_{2,\mathbf{q}}$.

According to the proof of theorem 2 in (Yarotsky, 2018), we can rewrite

$$\tilde{f}_{2,\mathbf{q}}(\mathbf{x}) = \lambda \sum_{\overline{\mathbf{m}}\in\left[-\frac{M}{N}+1,\ldots,\frac{M}{N}-1\right]^{d-1}} \sum_{m_1=-\frac{M}{N}+1}^{\frac{M}{N}-1} \widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}\left(m_1,\mathbf{x}\right) B_{\mathbf{q},\mathbf{n}}\left(m_1,\overline{\mathbf{m}}\right), \tag{21}$$

where

$$\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}\left(m_1,\mathbf{x}\right) = \begin{cases} \Phi_{\mathbf{n},\overline{\mathbf{m}}}\left(m_1,\mathbf{x}\right), & \mathbf{x}\in Q_{\mathbf{n}}, \mathbf{n}\in\left(\mathbf{q}+(3\mathbb{Z})^d\right)\cap[0,N]^d, \\ 0, & \text{otherwise}, \end{cases} \tag{22}$$

$$= \min\left(\sum_{s=-M/N+1}^{m_1} \phi\left(M\left(\mathbf{x}-\left(\Psi_{\mathbf{q}}(\mathbf{x})+\frac{(s,\overline{\mathbf{m}})}{M}\right)\right)\right), \theta_{\mathbf{q}}(\mathbf{x})\right), \tag{23}$$

$$\Phi_{\mathbf{n},\overline{\mathbf{m}}}\left(m_1,\mathbf{x}\right) = \sum_{s=-M/N+1}^{m_1} \phi\left(M\left(\mathbf{x}-\left(\frac{\mathbf{n}}{N}+\frac{(m_1,\overline{\mathbf{m}})}{M}\right)\right)\right), \tag{24}$$

$$\Psi_{\mathbf{q}}(\mathbf{x}) = \frac{\mathbf{n}}{N}, \quad \text{if } \mathbf{x}\in Q_{\mathbf{n}}, \mathbf{n}\in\left(\mathbf{q}+(3\mathbb{Z})^d\right)\cap[0,N]^d, \tag{25}$$

$$\text{further,} \quad \Psi_{\mathbf{q}}(\mathbf{x}) = \left(\psi_{q_1}\left(x_1\right),\ldots,\psi_{q_d}\left(x_d\right)\right), \tag{26}$$

$$\psi_q(x) = \frac{q}{N} + 3\sum_{k=0}^{\lceil N/3\rceil}\left(\left(x-\frac{q+3k+1}{N}\right)_+ - \left(x-\frac{q+3k+2}{N}\right)_+\right), \tag{27}$$

$$\theta_{\mathbf{q}}(\mathbf{x}) = N\sum_{\mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^d)\cap[0,N]^d}\left(1-\max_{s=1,\ldots,d}|Nx_s-n_s|\right)_+, \tag{28}$$

and

$$B_{\mathbf{q},\mathbf{n}}(\mathbf{m}) = A_{\mathbf{q},\mathbf{n}}\left(m_1,\overline{\mathbf{m}}\right) - A_{\mathbf{q},\mathbf{n}}\left(m_1+1,\overline{\mathbf{m}}\right), \quad \mathbf{m}\in\left[-\frac{M}{N}+1,\ldots,\frac{M}{N}-1\right]^d, \tag{29}$$

$$A_{\mathbf{q},\mathbf{n}}(\mathbf{m}) = \left\lfloor f_{2,\mathbf{q}}\left(\frac{\mathbf{n}}{N}+\frac{\mathbf{m}}{M}\right)/\lambda\right\rfloor, \quad \mathbf{m}\in\left[-\frac{M}{N},\ldots,\frac{M}{N}\right]^d. \tag{30}$$

$$\tag{31}$$

We skip many details for deriving these formulas and they can be found in (Yarotsky, 2018).

Moreover, the authors in (Yarotsky, 2018) introduces how to recover $b_{\mathbf{q},\mathbf{n}}$ from $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$. Since $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})\in\{-1,0,1\}$, we can encode all the $(2M/N-1)^d$ values $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ by a single ternary number

$$b_{\mathbf{q},\mathbf{n}} = \sum_{t=1}^{(2M/N-1)^d} 3^{-t}\left(B_{\mathbf{q},\mathbf{n}}\left(\mathbf{m}_t\right)+1\right), \tag{32}$$

where $t\mapsto\mathbf{m}_t$ is some enumeration of the multi-indices $\mathbf{m}$. The values $b_{\mathbf{q},\mathbf{n}}$ for all $\mathbf{q}$ and $\mathbf{n}$ will be stored as parameters in the network. Given $\mathbf{x}$, the relevant value of $b_{\mathbf{q},\mathbf{n}}$ can be selected among the values for all patches by the ReLU network computing

$$b_{\mathbf{q}}(\mathbf{x}) = \sum_{\mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^d)\cap[0,N]^d}\frac{b_{\mathbf{q},\mathbf{n}}}{2}\left((2-u)_+-(1-u)_+\right), \text{ where } u=\max_{s=1,\ldots,d}|Nx_s-n_s|. \tag{33}$$

If $\mathbf{x}$ belongs to some patch $Q_{\mathbf{n}}$, then $b_{\mathbf{q}}(\mathbf{x})=b_{\mathbf{q},\mathbf{n}}$, as required. If $\mathbf{x}$ does not belong to any cube $Q_{\mathbf{n}}$, $b_{\mathbf{q}}(\mathbf{x})$ will compute some unimportant value because that term will vanish according to (21,22). Then $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ can be reconstructed from $b_{\mathbf{q},\mathbf{n}}$ by the bit extraction methods.

In (Yarotsky, 2018), the author uses d channels to reserve the value of the input, another d channels to reserve the value of $\Psi_{\mathbf{q}}(\mathbf{x})$, and ten more channels to process some computations. Thus, it needs a ReLU network of width $2d+10$ to achieve the optimal approximation rates (20).

Now, we are ready to construct a network of width $d+11$ to generate ρ.

C.2. Construction Details

Now, given a continuous function f from $[0,1]^d$ to $\mathbb{R}$, we show that we can construct a register model in $\mathcal{I}_{\mathrm{ReLU}}\left(d, 10, 1; \mathcal{O}\left(L\right)\right)$ to generate ρ (Equation 18). In this register model, we allocate the first d channels, referred to as the source channels, for forwarding the input value $\mathbf{x}$, and the final channel, referred to as the collation channel, for storing and refreshing intermediate computations. For the sake of convenience in the narrative, we first suppose that our register model has a few more channels, that is $\mathcal{I}_{\mathrm{ReLU}}\left(d, 15, 1; \mathcal{O}\left(L\right)\right)$ where we have 15 channels for intermediate computations. We will call these 15 channels 'Channel*1*','Channel*2*',..., 'Channel*15*' respectively.

Stage 1. Computation of $\tilde{f}_1$.

Note that all of $\ell(\mathbf{x}) = 1+x_k, \ell(\mathbf{x}) = 1-x_k, \ell(\mathbf{x}) = 1+x_k-x_s$ are affine transformations. Then it follows from Equation (12) that $\phi(\mathbf{x})$ is a min string of length $\mathcal{O}(d^2)$. This allows us to use Channel*1* to compute $\phi(N\mathbf{x}-\mathbf{n})$. Then the value of $\tilde{f}_1$ can be stored in the collation channel. For each ϕ, we will consume not more than d^2 layers. By Equation (13), it will consume not more than d^2N^d layers to compute $\tilde{f}_1$. Moreover, Channel*1* becomes a garbage channel after computing $\tilde{f}_1$ so we can use it later.

Stage 2. Computation of $\theta_{\mathbf{q}}(x), b_{\mathbf{q}}(x)$.

We use three channels (Channel*2*-Channel*4*) for our compuations. Note that

$$u = \max_{s=1,\ldots,d} |Nx_s - n_s| = \max_{s=1,\ldots,d} \left\{Nx_s - n_s, -Nx_s + n_s\right\}.$$

Hecen, we then can use Channel*4* of depth $2d$ to compute u by Prop. B.6. Once we get the value of u for the first $\mathbf{n}$, we can compute $\frac{b_{\mathbf{q},\mathbf{n}}}{2}(2-u)_+$ and reserve it in Channel*2*. Then we use one layer in Channel*4* to compute $(1-u)_+$. Next, pass the value $-\frac{b_{\mathbf{q},\mathbf{n}}}{2}(1-u)_+$ to Channel*2* and add $\frac{b_{\mathbf{q},\mathbf{n}}}{2}(2-u)_+$ as the bias. At the same time, pass the value $N(1-u)_+$ to Channel*3* and reserve it. Now, Channel*2* and Channel*3* have reserved the value of the partial sum of $b_{\mathbf{q}}(\mathbf{x})$ and $\theta_{\mathbf{q}}(\mathbf{x})$ respectively. Then for any $\mathbf{n}$, we can first compute u and use the same process to compute $\frac{b_{\mathbf{q},\mathbf{n}}}{2}\left((2-u)_+ - (1-u)_+\right)$ and $N(1-u)_+$.By induction, we can finally compute $b_{\mathbf{q}}(\mathbf{x})$ and $\theta_{\mathbf{q}}(\mathbf{x})$ and reserve them in Channel*2* and Channel*3* respectively. The total process consumes not more than $2dN^d$ layers. Moreover, note that Channel*4* becomes a garbage channel, which we can use for other computations later.

Stage 3. Computation of $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ and $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ in parallel.

Substage 3.1. Reconstruction of $\{B_{\mathbf{q},\mathbf{n}}(\mathbf{m})\}$ from $b_{\mathbf{q},\mathbf{n}}$. (Yarotsky, 2018) has shown that this reconstruction process can be efficiently carried out using 4 channels. We state the idea of bit extraction. Let's consider the sequence z_t with $z_0 = b_{\mathbf{q},\mathbf{n}}$ and $z_{t+1} = 3z_t - \lfloor 3z_t \rfloor$. It follows that the bit $B_{\mathbf{q},\mathbf{n}}\left(\mathbf{m}_t\right)$ is give by $B_{\mathbf{q},\mathbf{n}}\left(\mathbf{m}_t\right) = \lfloor 3z_{t-1} \rfloor - 1$ for all t. For the implementation of these computations by a ReLU network, it is necessary to compute $\lfloor 3z_t \rfloor$ for all z_t. This can be achieved by a piecewise linear function $\chi_\epsilon : [0,3) \to \mathbb{R}$ defined as

$$\chi_\epsilon(x) = \begin{cases} 0, & x \in [0, 1-\epsilon] \\ 1, & x \in [1, 2-\epsilon], \\ 2, & x \in [2, 3-\epsilon] \end{cases} \tag{34}$$

Such a function can be realized by $\chi_\epsilon(x) = \frac{1}{\epsilon}(x-(1-\epsilon))_+ - \frac{1}{\epsilon}(x-1)_+ + \frac{1}{\epsilon}(x-(2-\epsilon))_+ - \frac{1}{\epsilon}(x-2)_+$. It is important to note that if $\epsilon < 3^{-(2M/N-1)^d}$, then for all t the number $3z_t$ will fall within one of the three intervals in the right-hand side of Equation (34) and hence $\chi_\epsilon\left(3z_t\right) = \lfloor 3z_t \rfloor$.

Thus, using four channels (Channel*5*-Channel*8*), we can decode $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ from $b_{\mathbf{q},\mathbf{n}}$. Channel*5* is used to store and refresh the value of z_t. Channel*6* is used to store and refresh $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$. Channel*7* and Channel*8* are used to compute $\chi_\epsilon(3z_t)$. Hence we can reconstruct the values $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ for all $\mathbf{m}$ by a ReLU network with not more than $4(M/N)^d$ layers. For each $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$, we only need $O(1)$ layers to decode it.

Substage 3.2. Computation of $\widetilde{\Phi}_{\overline{\mathbf{m}}}$. Unlike (Yarotsky, 2018), we do not compute and reserve $\Phi_{\mathbf{q}}(\mathbf{x})$ in advance because we need to conserve channels. In this process, we assume our budget has 6 channels (Channel*9*-Channel*14*).

Note that by formula (27), we can compute and refresh $\psi_{q_i}(x_i)$ and $\psi_{q_j}(x_j)$ $(i \neq j)$ using two channels (Channel*9* and Channel*10* respectively). In this process, Channel*11* can be used to compute some intermediate value. Concretely, Channel*9* first compute $3(x_i - \frac{q_i+1}{N})_+$ and Channel*11* compute $3(x_i - \frac{q_i+2}{N})_+$. Then pass the value $-3(x_i - \frac{q_i+2}{N})_+$ to Channel*9* and add the pre-value $3(x_i - \frac{q_i+1}{N})_+$ as bias. Then we have computed one term of the sum (27). With the same process and by induction, we can consume not more than $2N$ layer to compute $\psi_{q_i}(x_i)$ and $\psi_{q_j}(x_j)$.

Now, suppose we have got the value y_i and y_j in Channel*9* and Channel*10* respectively where $y_i := M(x_i - \psi_{q_i}(x_i) - m_i/M)$. Note that for any real number a, b, $(\min\{a, b\})_+ = \min\{a_+, b_+\}$. We then use Channel*11* to compute the current value $\min\{1 + y_i, 1 - y_i, 1 - y_i + y_j, 1 - y_j + y_i, \phi'\}$ where ϕ' is the partial min string of $\phi(\mathbf{y})$ for $\mathbf{y} = (y_1, \cdots, y_d)$. Now, this value will be passed to Channel*12* for refreshing. We iterate i from 1 to d, and for each i we iterate j from i to d, repeating this process. Finally, we can compute $\phi(\mathbf{y})$ in Channel*12*. Moreover, The remaining two channels are used to store the partial sum $\sum_s^{m_1} \phi$ (Channel*13*) and compute $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ (Channel*14*) by (22), respectively. For each s, we will consume not more than $2d^2N$ layers. Because the partial sum $\sum_s^{m_1} \phi$ is reserved, we will consume not more than $2d^2N$ layers for each m_1 from $M/N - 1$ to $M/N + 1$.

Stage 4. Computation of the multiplication of $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ and $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$.

For any $a \in [0, 1], b \in \{-1, 0, 1\}$ we have $ab = (a + b - 1)_+ + (-a - b)_+ - (-b)_+$. We can find $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ and $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ in the same layer due to the parallel computation of $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ and $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$. Then we only need one channel (Channel*15*) for this multiplication operation. To show this, let's $a = \widetilde{\Phi}_{\overline{\mathbf{m}}}$ and $b = B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$. We can compute $(a + b - 1)_+$ first in Channel*15* and pass the value $\lambda(a + b - 1)_+$ to the collation channel. Then compute $(-a - b)_+$, $(-b)_+$ and pass the value $\lambda(-a - b)_+$, $-\lambda(-b)_+$ to collation channel. Then we have reserved one term of (21) in the collation channel. Then by repeating this process, we can compute and reserve $\tilde{f}_{2,\mathbf{q}}$ in collation channel with not more than $2dN^d + 2dN \times (\frac{M}{N})^d$ layers.

Stage 5. For each $\mathbf{q} \in \{0, 1, 2\}^d$, we repeat the stage 2-4. Finally, we will cost $3^d \left(2dN^d + 2dN \times (\frac{M}{N})^d\right)$ layers to generate $\tilde{f}_2 = \sum_{\mathbf{q}\in\{0,1,2\}^d} \tilde{f}_{2,\mathbf{q}}$.

Reduction of the number of channels. Note that Channel*1* and Channel*4* are idle after stage 1 and stage 2. Therefore, in substage 3.1, the tasks of Channel*7* and Channel*8* can be accomplished by Channel*1* and Channel*4*, respectively. Although the computations of $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ and $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ are in parallel, for each $\mathbf{q}, \mathbf{n}, \mathbf{m}$, they can be computed sequentially. We only need to find $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ and $\widetilde{\Phi}_{\overline{\mathbf{m}}}$ in the same layer. Thus, the tasks of channel 9 and Channel*8* 10 can also be accomplished by Channel*1* and Channel*4*. Moreover, the task of Channel*15* can also be accomplished by Channel*11* because after stage 3 Channel*11* will become idle. In the end, we can save the use of five channels, i.e., our register model belongs to $\mathcal{I}_{\mathrm{ReLU}}(d, 15, 1; \hat{L})$ where $\hat{L}$ is the depth we need.

Approximation rate. Recall that $N = c_1 W^{1/d}$, $M = c_2 W^{2/d}$. We can set $M = L^{2/d}$ so that by Equation (19) we have

$$\|f - \rho\|_{L^\infty} \leq 3^d(3d + 1)\left(6d^{3/2} + 1\right)\omega_f\left(L^{-\frac{2}{d}}\right). \tag{35}$$

To generate ρ, we will totally cost

$$d^2N^d + 3^d\left(2dN^d + 2dN \times (\frac{M}{N})^d\right)$$

layers. Let $c_2 \leq c_1^2$ and c_1 be small enough. Then the number of layers is not more than $6d3^dL^{1+1/d}$. By letting $\tilde{L} = L^{1+1/d}$, we can get the approximation rate

$$\|f - \rho\|_{L^\infty} \leq 3^d(3d + 1)\left(6d^{3/2} + 1\right)\omega_f\left(\tilde{L}^{-\frac{2}{d+1}}\right) \tag{36}$$

by a narrow ReLU network with width $d + 11$ and depth $\mathcal{O}(\tilde{L}) = 6d3^d\tilde{L}$.

C.3. Approximation of Mappings: Proof of (ii) of Theorem 3.1

Now, we assume $\boldsymbol{f} = (f_1, f_2, \cdots, f_v)$ is a continuous mapping in from $[0,1]^d$ to $\mathbb{R}^v$. It follows from Section C.2 we have a register model $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_v) \in \mathcal{I}^{d,v}_{\mathrm{ReLU}}(d, 10, v; 6d3^d vL)$ such that for $\mathbf{x} \in [0,1]^d$

$$\|f_i - \rho_i\|_{L^\infty([0,1]^d)} \leq 3^d(3d+1)\left(6d^{3/2}+1\right)\omega_f\left(L^{-\frac{2}{d+1}}\right). \tag{37}$$

Note that $\mathcal{I}^{d,v}_{\mathrm{ReLU}}(d, 10, v; 6d3^d vL) \subset \mathcal{NN}^{d,v}_{\mathrm{ReLU}}(d+v+10, 6d3^d vL)$ by Lemma A.2. Then by Lemma A.3 and A.4, we end the proof of (ii) of Theorem 3.1.

D. Diverse Activation Functions

D.1. Definitions of Diverse Activation Functions in the Set Σ

- $\mathrm{ReLU}(x) = \max\{0, x\}$.
- $\mathrm{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0 \end{cases}$ where $\alpha \in (0,1)$.
- $\mathrm{ReLU}^2(x) = \max\{0, x\}^2$.
- Standard Sigmoid:
$$\mathrm{Sigmoid}(x) = \frac{1}{1+e^{-x}}.$$
- Tanh Function:
$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$
- $\mathrm{Arctan}(x)$:
$$\arctan(x) = \tan^{-1}(x) : \mathbb{R} \to (-\frac{\pi}{2}, \frac{\pi}{2})$$
- Softsign:
$$\mathrm{Softsign}(x) = \frac{x}{1+|x|}.$$
- Derivative of SiLU (dSiLU):
$$\mathrm{dSiLU}(x) = \frac{1 + e^{-x} + xe^{-x}}{(1+e^{-x})^2}.$$
- Soft-Root-Sign (SRS):
$$\mathrm{SRS}(x) = \frac{x}{x/\alpha + e^{-x/\beta}} \quad \text{with } \alpha, \beta \in (0, \infty).$$
- Exponential linear unit (ELU):
$$\mathrm{ELU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha\left(e^x - 1\right) & \text{for } x < 0 \end{cases} \quad \text{with} \quad \alpha \in \mathbb{R}.$$
- Scaled Exponential Linear Unit (SELU): for $\lambda \in (0, \infty)$ and $\alpha \in \mathbb{R}$,
$$\mathrm{SELU}(x) = \lambda \begin{cases} x & \text{for } x \geq 0 \\ \alpha\left(e^x - 1\right) & \text{for } x < 0 \end{cases}$$
- $\mathrm{Softplus}(x) = \ln\left(1 + e^x\right)$.

- Sigmoid Linear Unit (SiLU):
$$\mathrm{SiLU}(x) = \frac{x}{1+e^{-x}}$$
.
- $\mathrm{Swish}(x) = \frac{x}{1+e^{-\beta x}}$ with $\beta \in (0, \infty)$.
- $\mathrm{Mish}(x) = x \cdot \mathrm{Tanh}(\mathrm{Softplus}(x))$.
- Gaussian Error Linear Unit (GELU):
$$\mathrm{GELU}(x) = x \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt \quad \text{with } \mu \in \mathbb{R} \text{ and } \sigma \in (0, \infty).$$

D.2. Proof of Corollary 3.2

In this section, we extend our results to networks equipped with other activation functions. Recently, (Zhang et al., 2023) has investigated the relationship between ReLU networks and networks with diverse activation functions. For the sake of convenience in our discussion, we categorize commonly used activation functions into the following sets according to (Zhang et al., 2023).

(i) Piecewise smooth function set $\mathscr{A}_{1,k}$ where k is the order of smoothness, is defined as

$$\mathscr{A}_{1,k} := \{\sigma : \mathbb{R} \to \mathbb{R} \,|\, \exists a_0, b_0 \in \mathbb{R}, a_0 < b_0, \sigma \in C^k\left((a_0, b_0)\right), \\ \exists x_0 \in (a_0, b_0), \mathbb{R} \ni \lim_{t \to 0^-} \frac{\sigma^{(k)}(x_0 + t) - \sigma^{(k)}(x_0)}{t} \neq \lim_{t \to 0^+} \frac{\sigma^{(k)}(x_0 + t) - \sigma^{(k)}(x_0)}{t} \in \mathbb{R}\Big\}.$$

The following common used activation functions are in $\mathscr{A}_{1,k}$:

- $\mathrm{ReLU}, \mathrm{LeakyReLU} \in \mathscr{A}_{1,0}$.
- $\mathrm{ReLU}^2 \in \mathscr{A}_{1,1}$.

(ii) $\widetilde{\mathscr{A}_2}$ is a specific subset of $\mathscr{A}_2$. It require that either $\lim_{x \to -\infty} h(x)$ or $\lim_{x \to \infty} h(x)$ must be equal to 0, and concretely defined as

$$\widetilde{\mathscr{A}_2} := \{\sigma : \mathbb{R} \to \mathbb{R} \,|\, \forall x \in \mathbb{R}, \sigma(x) := (x + b_0) \cdot h(x) + b_1, \quad b_0, b_1 \in \mathbb{R}, \quad h : \mathbb{R} \to \mathbb{R}, \\ \sup_{x \in \mathbb{R}} |h(x)| < \infty, \quad \mathbb{R} \ni L_1 = \lim_{x \to -\infty} h(x) \neq \lim_{x \to \infty} h(x) = L_2 \in \mathbb{R}, \quad L_1 \cdot L_2 = 0\Big\}.$$

The following activation functions are in $\widetilde{\mathscr{A}_2}$:

- $\mathrm{Softplus}$, $\mathrm{GELU}(\mu = 0, \sigma > 0)$, SiLU, $\mathrm{Swish}(\beta > 0)$, Mish, $\mathrm{ELU}(\alpha > 0)$, SELU.
- $\varrho_1(x) = x \cdot \mathrm{dSiLU}(x)$.
- $\varrho_2(x) = x \cdot (\mathrm{Softsign}(x)/2 + 1/2)$.
- $\varrho_3(x) = x \cdot (\mathrm{Arctan}(x)/\pi + 1/2)$.

(iii) $\mathscr{A}_3$ consists of functions with the similar shape of Sigmoid, defined as

$$\mathscr{A}_3 := \left\{\sigma : \mathbb{R} \to \mathbb{R} \mid \sup_{x \in \mathbb{R}} |\sigma(x)| < \infty, \quad \exists x_0 \in \mathbb{R}, \sigma''(x_0) \neq 0, \mathbb{R} \ni \lim_{x \to -\infty} \sigma(x) \neq \lim_{x \to \infty} \sigma(x) \in \mathbb{R}\right\}.$$

The following activation functions are in $\mathscr{A}_3$:

- Sigmoid, Tanh, Arctan, Softsign, dSiLU, SRS.

Corollary 3.2 is the direct result of the following theorem.

Theorem D.1 ((Zhang et al., 2023)). *Suppose $\sigma \in \mathscr{A}$ and $\phi_{\mathrm{ReLU}} \in \mathcal{NN}^{d,v}_{\mathrm{ReLU}}(N, L)$ with $N, L, d, v \in \mathbb{N}^+$. Then for any $\varepsilon > 0$ and $A > 0$, there exists $\phi_\sigma \in \mathcal{NN}^{d,v}_{\sigma}(\widetilde{N}, \widetilde{L})$ such that*

$$\|\phi_\sigma - \phi_{\mathrm{ReLU}}\|_{L^\infty([-A,A]^d)} < \varepsilon$$

where i) $\widetilde{N} = k + 2, \widetilde{L} = L$ if $\mathscr{A} = \mathscr{A}_{1,k}$, ii) $\widetilde{N} = N, \widetilde{L} = L$ if $\mathscr{A} = \widetilde{\mathscr{A}_2}$ and iii) $\widetilde{N} = 3N, \widetilde{L} = 2L$ if $\mathscr{A} = \mathscr{A}_3$.

Let ϕ_σ be a neural network equipped with activation σ of fixed width and depth $\mathcal{O}(L)$. For any given continuous function or mapping $\boldsymbol{f}$ over $[0,1]^d$, we have

$$\|\phi_\sigma - f\| \le \|f - \phi_{\mathrm{ReLU}}\| + \|\phi_{\mathrm{ReLU}} - \phi_\sigma\|$$

where $\|\boldsymbol{f} - \phi_{\mathrm{ReLU}}\| = \mathcal{O}(\omega_f(L^{-2/d}))$ by Theorem 1.1 or 3.1 and $\|\phi_{\mathrm{ReLU}} - \phi_\sigma\| = \mathcal{O}(\omega_f(L^{-2/d}))$ by letting $\varepsilon = \mathcal{O}(L^{-2/d})$ in Theorem D.1. Thus, ϕ_σ can achieve the rate $\mathcal{O}(\omega_f(L^{-2/d}))$ for approximating a given continuous function f over $[0,1]^d$. Note that the norm here can be L^p for $p \in [1, \infty]$.

D.3. Proof of Theorem 3.5

For L^∞ norm, we use the following theorem.

Theorem D.2 ((Yarotsky, 2017; Shen et al., 2022b)). *Assume $\mathcal{F}$ is a set of functions mapping from $[0,1]^d$ to $\mathbb{R}$. For any $\varepsilon > 0$, if $\mathrm{VCDim}(\mathcal{F}) \ge 1$ and*

$$\inf_{\phi \in \mathcal{F}} \|\phi - f\|_{L^\infty([0,1]^d)} \le \varepsilon, \quad \text{for any } f \in \mathrm{Lip}\left([0,1]^d\right),$$

then $\mathrm{VCDim}(\mathcal{F}) \ge (9\varepsilon)^{-d}$.

Hence, if we let $\varepsilon = \mathcal{E}(d, L) := \sup_{f \in \mathrm{Lip}([0,1]^d, \mu)} \left(\inf_{\rho \in \mathcal{H}_\sigma(L)} \|\rho - f\|_{L^p([0,1]^d)}\right)$ and $\mathcal{F} = \mathcal{H}_\sigma(L)$, we have the conclusion.

For L^p norm where $1 \le p < \infty$, a recent result works.

Theorem D.3 ((Siegel, 2023)). *Let $p > 0, \Omega = [0,1]^d$ and suppose that K is a translation invariant class of functions whose VC-dimension is at most n. By translation invariant we mean that $f \in K$ implies that $f(\cdot - v) \in K$ for any fixed vector $v \in \mathbb{R}^d$. Then there exists an $f \in \mathrm{Lip}([0,1]^d)$ such that*

$$\inf_{g \in K} \|f - g\|_{L^p(\Omega)} \ge C(p,d) n^{-\frac{1}{d}} \|f\|_{\sup[0,1]^d}$$

Note that the set of networks with any activation σ is a translation invariant class. We get the conclusion.

D.4. Proof of Corollary 3.6

It suffices to consider the VC dimension of a set consisting of networks with fixed width and some activation function σ.

For (i) of Corollary 3.6, we use the following theorem.

Theorem D.4 (Theorem 8.4 (Anthony et al., 1999)). *Suppose h is a function from $\mathbb{R}^d \times \mathbb{R}^W$ to $\{0,1\}$ and let*

$$H = \left\{x \mapsto h(x,a) : a \in \mathbb{R}^W\right\}$$

be the class determined by h. Suppose that h can be computed by an algorithm that takes as input the pair $(x,a) \in \mathbb{R}^d \times \mathbb{R}^W$ and returns $h(x,a)$ after no more than t operations of the following types:

- *the arithmetic operations, $+ - x$, and $/$ on real numbers,*

- *jumps conditioned on* $>, \geq, <, \leq, =$*, and* $\neq$ *comparisons of real numbers, and*
- *output 0 or 1.*

Then VCdim $(H) \leq 4W(t+2)$.

If σ is a piecewise polynomial function, the time to compute each σ is $\mathcal{O}(1)$ where the constant is related to the number of pieces and the degree of the polynomial. Here note that we can use the operation 2 in the above theorem to compute which piece x belongs to. For a narrow network with width $\mathcal{O}(L)$, the parameters $W \propto L$. Thus, we will take $\mathcal{O}(L)$ time in total to compute a narrow σ network with width $\mathcal{O}(L)$. From the above theorem, its VC dimension is $\mathcal{O}(L^2)$. Moreover, for $\mathrm{Softsign}$ activation, one can also use the above theorem to compute the VC dimension of $\mathcal{H}_{\mathrm{Softsign}}(L)$ is $\mathcal{O}(L^2)$.

Theorem D.5 (Theorem 8.14 (Anthony et al., 1999))**.** *Let* h *be a function from* $\mathbb{R}^d \times \mathbb{R}^W$ *to* $\{0,1\}$*, determining the class*

$$H = \left\{x \mapsto h(x,a) : a \in \mathbb{R}^W\right\}.$$

Suppose that h *can be computed by an algorithm that takes as input the pair* $(x,a) \in \mathbb{R}^d \times \mathbb{R}^W$ *and returns* $h(x,a)$ *after no more than* t *of the following operations:*

- *the exponential function* $\alpha \mapsto e^\alpha$ *on real numbers,*
- *the arithmetic operations,,*$+ - x$*, and / on real numbers,*
- *jumps conditioned on* $>, \geq, <, \leq, =$*, and* $\neq$ *comparisons of real numbers, and*
- *output 0 or 1.*

Then $\mathrm{VCdim}(H) \leq t^2 W\left(W + 19\log_2(9W)\right)$.

For σ belonging to $\{\mathrm{ELU}, \mathrm{SELU}, \mathrm{SiLU}, \mathrm{Swish}, \mathrm{Mish}, \mathrm{Sigmoid}, \mathrm{Tanh}, \mathrm{dSiLU}, \mathrm{SRS}\}$, the above theorem can directly show $\mathrm{VCdim}(H_\sigma(L)) = \mathcal{O}(L^4)$.

Besides, note the above theorem can not apply to Arctan activation. But from another work (Karpinski & Macintyre, 1997), we could get $\mathrm{VCdim}(H_\sigma(L)) = \mathcal{O}(L^4)$ if σ is Arctan. Let's first introduce the definitions of Pfaffian functions.

Definition D.6 ((Yarotsky, 2021))**.** A Pfaffian chain is a sequence $f_1, \ldots, f_\ell$ of real analytic functions defined on a common connected domain $U \subset \mathbb{R}^d$ and such that the equations

$$\frac{\partial f_i}{\partial x_j}(\mathbf{x}) = P_{ij}\left(\mathbf{x}, f_1(\mathbf{x}), \ldots, f_i(\mathbf{x})\right), \quad \text{for } 1 \leq i \leq \ell, 1 \leq j \leq d$$

hold in U for some polynomials P_{ij}. A Pfaffian function in the chain $(f_1, \ldots, f_\ell)$ is a function on U that can be expressed as a polynomial P in the variables $(\mathbf{x}, f_1(\mathbf{x}), \ldots, f_\ell(\mathbf{x}))$.

It follows from (Karpinski & Macintyre, 1997) that if σ is Pfaffian, then the VC dimension of $\mathcal{H}_\sigma(L)$ is $\mathcal{O}(L^4)$. Note that $f(x) = \arctan(x)$, is a Pfaffian function. To see this, take $f_1(x) = 1/\left(1+x^2\right)$ and $f_2(x) = \arctan(x)$; then $f_1'(x) = -2x/\left(1+x^2\right)^2 = -2x f_1(x)^2$, and $f_2'(x) = 1/\left(1+x^2\right) = f_1(x)$. Thus, we end the proof.