On Enhancing Expressive Power via Compositions of Single Fixed-Size ReLU Network

Shijun Zhang [1] **Jianfeng Lu** [1] **Hongkai Zhao** [1]

Abstract

This paper explores the expressive power of deep neural networks through the framework of function compositions. We demonstrate that the repeated compositions of a single fixed-size ReLU network exhibit surprising expressive power, despite the limited expressive capabilities of the individual network itself. Specifically, we prove by construction that $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ can approximate 1-Lipschitz continuous functions on $[0,1]^d$ with an error $\mathcal{O}(r^{-1/d})$, where $\boldsymbol{g}$ is realized by a fixed-size ReLU network, $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are two affine linear maps matching the dimensions, and $\boldsymbol{g}^{\circ r}$ denotes the r-times composition of $\boldsymbol{g}$. Furthermore, we extend such a result to generic continuous functions on $[0,1]^d$ with the approximation error characterized by the modulus of continuity. Our results reveal that a continuous-depth network generated via a dynamical system has immense approximation power even if its dynamics function is time-independent and realized by a fixed-size ReLU network.

1. Introduction

In recent years, there has been a notable increase in the exploration of the expressive power of deep neural networks, driven by their impressive success in various learning tasks. The increasing size of deep neural network models poses significant challenges in terms of training and computational requirements. Consequently, numerous techniques have emerged to compress and expedite these models, with the goal of alleviating the associated computational complexity. These techniques predominantly center around parameter-sharing schemes, which efficiently reduce the number of parameters, leading to reductions in memory usage and computation costs.

[1]Department of Mathematics, Duke University, USA. Correspondence to: Shijun Zhang <shijun.zhang@duke.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

This paper explores the expressive power of deep neural networks, approaching it from the standpoint of function compositions. We focus on a novel network architecture constructed through the repeated compositions of a single fixed-size network, enabling parameter sharing. To illustrate our ideas and concepts, we specifically utilize the rectified linear unit (ReLU) activation function. Our investigation reveals that the repeated compositions of a single fixed-size ReLU network demonstrate surprising expressive power, even though the individual network itself has limited expressive capabilities. These findings provide new insights into the potential of parameter-sharing schemes in neural networks, showcasing their ability to reduce computational complexity while preserving a high level of expressive power.

For ease of notation, we employ $\mathcal{NN}\{N,\, L;\, \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}\}$ to represent the set of functions $\boldsymbol{\phi} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ that can be realized by ReLU networks of width $N \in \mathbb{N}^+$ and depth $L \in \mathbb{N}^+$. In our context, the width of a network means the maximum number of neurons in a hidden layer and the depth refers to the number of hidden layers. Let $\boldsymbol{g}^{\circ r}$ denote the r-times composition of $\boldsymbol{g}$, e.g., $\boldsymbol{g}^{\circ 3} = \boldsymbol{g} \circ \boldsymbol{g} \circ \boldsymbol{g}$. In the degenerate case, $\boldsymbol{g}^{\circ 0}$ represents the identity map. We use $C([0,1]^d)$ to denote the set of continuous functions on $[0,1]^d$ and define the modulus of continuity of a continuous function $f \in C([0,1]^d)$ via

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

for any $t \ge 0$. Under these settings, we can construct $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ to approximate a continuous function $f \in C([0,1]^d)$ with an error $\mathcal{O}\big(\omega_f(r^{-1/d})\big)$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are two affine linear maps and $\boldsymbol{g}$ is realized by a fixed-size ReLU network, as shown in the theorem below.

Theorem 1.1. *Given any $f \in C([0,1]^d)$, $r \in \mathbb{N}^+$, and $p \in [1,\infty)$, there exist $\boldsymbol{g} \in \mathcal{NN}\{69d+48,\, 5;\, \mathbb{R}^{5d+5} \to \mathbb{R}^{5d+5}\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+5}$ and $\mathcal{L}_2 : \mathbb{R}^{5d+5} \to \mathbb{R}$ such that*

$$\left\|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1 - f\right\|_{L^p([0,1]^d)} \le 6\sqrt{d}\,\omega_f(r^{-1/d}).$$

It should be noted that in Theorem 1.1, the affine linear maps $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are used to ensure matching dimensions and can be replaced by various other functions that

achieve the desired input and output dimensions. In our research, we choose a straightforward approach by considering them as affine linear maps. In Theorem 1.1, we propose a novel network architecture constructed via repeated compositions of a single sub-network, which will be referred to as repeated-composition networks (RCNets). The hypothesis space of the RCNet corresponding to $\boldsymbol{g}$ is defined as

$$\mathcal{H}(\boldsymbol{g}) \coloneqq \Big\{\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1 : r \in \mathbb{N},\ \boldsymbol{\mathcal{L}}_1 \text{ and } \mathcal{L}_2 \text{ are affine}\Big\}.$$

Then we have an immediate corollary as follows.

Corollary 1.2. *Given any $p \in [1, \infty)$, suppose $\mathcal{H}(\boldsymbol{g})$ is defined as mentioned above and set $\boldsymbol{\mathcal{G}} = \mathcal{NN}\{69d + 48,\ 5;\ \mathbb{R}^{5d+5} \to \mathbb{R}^{5d+5}\}$. Then $\mathcal{H} = \cup_{\boldsymbol{g} \in \boldsymbol{\mathcal{G}}} \mathcal{H}(\boldsymbol{g})$ is dense in $L^p([0,1]^d)$ in terms of the L^p-norm.*

The proof of Corollary 1.2 is straightforward. Theorem 1.1 implies $\mathcal{H}$ is dense in $C([0,1]^d)$ in terms of the L^p-norm for any $p \in [1, \infty)$. Recall that $C([0,1]^d)$ is dense in the Lebesgue spaces $L^p([0,1]^d)$ for any $p \in [1, \infty)$. Therefore, we can conclude that $\mathcal{H}$ is dense in $L^p([0,1]^d)$ in terms of the L^p-norm for any $p \in [1, \infty)$. Furthermore, it should be noted that the set $\boldsymbol{\mathcal{G}}$ in Corollary 1.2 is generated by a fixed-size ReLU network. As a result, $\boldsymbol{\mathcal{G}}$ is a set of continuous piecewise linear functions with (at most) a fixed number of pieces.

It is important to note that the approximation error in Theorem 1.1 is quantified by the L^p-norm for any $p \in [1, \infty)$. However, it is possible to extend this result to the L^∞-norm as well, although the associated constants will be significantly larger.

Theorem 1.3. *Given any $f \in C([0,1]^d)$ and $r \in \mathbb{N}^+$, there exist $\boldsymbol{g} \in \mathcal{NN}\{4^{d+5}d,\ 3 + 2d;\ \mathbb{R}^{\widetilde{d}} \to \mathbb{R}^{\widetilde{d}}\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{\widetilde{d}}$ and $\mathcal{L}_2 : \mathbb{R}^{\widetilde{d}} \to \mathbb{R}$ such that*

$$\big|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+2d-1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) - f(\boldsymbol{x})\big| \le 6\sqrt{d}\,\omega_f(r^{-1/d})$$

for any $\boldsymbol{x} \in [0,1]^d$, where $\widetilde{d} = 3^d(5d+4) - 1$.

The main ideas for proving Theorems 1.1 and 1.3 are provided in Section 3 and the detailed proofs of these two theorems can be found in Section A of the appendix.

In general, it is challenging to simplify the approximation error in Theorem 1.1 (or 1.3) due to the complexity of $\omega_f(\cdot)$. However, in the case of special target function spaces like Hölder continuous function space, one can simplify the approximation error to make its dependence on r explicit. If f is an Hölder continuous function on $[0,1]^d$ of order $\alpha \in (0,1]$ with an Hölder constant $\lambda > 0$, we have

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \le \lambda \|\boldsymbol{x} - \boldsymbol{y}\|_2^\alpha \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in [0,1]^d,$$

implying $\omega_f(t) \le \lambda t^\alpha$ for any $t \ge 0$. Thus, the approximation error in Theorem 1.1 (or 1.3) can be simplified to $6\lambda\sqrt{d}\,r^{-\alpha/d}$. In the special case of $\alpha = 1$, where f is a Lipschitz continuous function with a Lipschitz constant $\lambda > 0$, the approximation error can be further simplified to $6\lambda\sqrt{d}\,r^{-1/d}$.

A constant-width ReLU network of depth $\mathcal{O}(r)$ can be represented as $\mathcal{L}_2 \circ \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_2 \circ \boldsymbol{g}_1 \circ \boldsymbol{\mathcal{L}}_1$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are affine linear maps and each $\boldsymbol{g}_i$ is a fixed-size ReLU network. It has been shown in (Shen et al., 2020; Yarotsky, 2018; Zhang, 2020) that the optimal approximation error is $\mathcal{O}(r^{-2/d})$ when using $\mathcal{L}_2 \circ \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_2 \circ \boldsymbol{g}_1 \circ \boldsymbol{\mathcal{L}}_1$ to approximate 1-Lipschitz continuous functions on $[0,1]^d$. In contrast, our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ can approximate 1-Lipschitz continuous functions on $[0,1]^d$ with an error $\mathcal{O}(r^{-1/d})$, where $\boldsymbol{g}$ is a fixed-size ReLU network. That means, at a price of a slightly worse approximation error, our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ essentially shares most of the parameters in $\mathcal{L}_2 \circ \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_2 \circ \boldsymbol{g}_1 \circ \boldsymbol{\mathcal{L}}_1$ and reduce trainable parameters to a constant. Furthermore, our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ is anticipated to exhibit improved gradient behavior compared to $\mathcal{L}_2 \circ \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_2 \circ \boldsymbol{g}_1 \circ \boldsymbol{\mathcal{L}}_1$ as the gradient with respect to the parameters in $\boldsymbol{g}$ is less likely to vanish for larger values of r.

Next, we point out some relations between our approximation results and dynamical systems. Our results reveal that a continuous-depth network generated via a dynamical system has enormous approximation power even if the dynamics is time-invariant and realized by a fixed-size ReLU network. Let us now delve into further details regarding this matter. A dynamical system is generally described by an ordinary differential equation (ODE)

$$\tfrac{d}{dt}\boldsymbol{z}(t) = \boldsymbol{F}\big(\boldsymbol{z}(t), t, \boldsymbol{\theta}\big), \quad t \in [0,T], \quad \boldsymbol{z}(0) = \boldsymbol{z}_0, \quad (1)$$

where $\boldsymbol{F} : \mathbb{R}^{n+1} \times \boldsymbol{\Theta} \to \mathbb{R}^n$ is the dynamics function of this dynamical system, parameterized with $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ is the parameter space.

For any $\boldsymbol{y} = \boldsymbol{z}_0 \in \mathbb{R}^n$, $\boldsymbol{z}(T)$ can be regarded as a function of $\boldsymbol{y}$ and we denote this function by $\boldsymbol{\Phi}(\cdot, \boldsymbol{\theta}) : \mathbb{R}^n \to \mathbb{R}^n$. Such a map is known as the flow map (or Poincaré map) of the dynamical system (1). Then we can use $\mathcal{L}_2 \circ \boldsymbol{\Phi}(\cdot, \boldsymbol{\theta}) \circ \boldsymbol{\mathcal{L}}_1$ to approximate a given target function $f : \mathbb{R}^d \to \mathbb{R}$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are two affine linear maps matching the dimensions.

Choose a large $S \in \mathbb{N}^+$ and set $\delta = T/S$. It follows from ODE (1) that

$$\boldsymbol{z}\big(\delta(s+1)\big) = \boldsymbol{z}(\delta s) + \int_{\delta s}^{\delta(s+1)} \boldsymbol{F}\big(\boldsymbol{z}(t), t, \boldsymbol{\theta}\big)dt$$

for $s = 0, 1, \cdots, S-1$. We denote $\boldsymbol{z}_s$ as the numerical solution and use it to approximate the true solution $\boldsymbol{z}(\delta s)$ for $s = 0, 1, \cdots, S$. By using the forward Euler method to

discretize ODE (1), we have

$$\boldsymbol{z}_{s+1} = \boldsymbol{z}_s + \delta \boldsymbol{F}\big(\boldsymbol{z}_s, \delta s, \boldsymbol{\theta}\big)$$

for $s = 0, 1, \cdots, S-1$. Such an iteration step can be regarded as a residual network (He et al., 2016). Thus, a dynamical system can be viewed as a continuous-time version of a residual network. The network generated via a dynamical system is generally called continuous-depth network. The function $\mathcal{L}_2 \circ \boldsymbol{\Phi}(\cdot, \boldsymbol{\theta}) \circ \boldsymbol{\mathcal{L}}_1$ mentioned above is indeed generated by a continuous-depth network. As we know, $\boldsymbol{z}_S$ can approximate $\boldsymbol{z}(\delta S) = \boldsymbol{z}(T)$ arbitrarily well for sufficiently large S with some proper conditions on the dynamics function $\boldsymbol{F}$.

Suppose $\boldsymbol{F}(\boldsymbol{y}, t, \boldsymbol{\theta})$ given in ODE (1) is independent of t for any $(\boldsymbol{y}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \boldsymbol{\Theta}$. Define $\boldsymbol{g}_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}^n$ via

$$\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{y}) \coloneqq \boldsymbol{y} + \delta \boldsymbol{F}(\boldsymbol{y}, 0, \boldsymbol{\theta}).$$

Then, we have

$$\begin{aligned} \boldsymbol{z}_{s+1} &= \boldsymbol{z}_s + \delta \boldsymbol{F}\big(\boldsymbol{z}_s, \delta s, \boldsymbol{\theta}\big) \\ &= \boldsymbol{z}_s + \delta \boldsymbol{F}\big(\boldsymbol{z}_s, 0, \boldsymbol{\theta}\big) = \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}_s) \end{aligned}$$

for $s = 0, 1, \cdots, S-1$, implying $\boldsymbol{z}_S = \boldsymbol{g}_{\boldsymbol{\theta}}^{\circ S}(\boldsymbol{z}_0)$. It follows that, for any $\boldsymbol{y} = \boldsymbol{z}_0 \in \mathbb{R}^n$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, we have

$$\begin{aligned} \boldsymbol{g}_{\boldsymbol{\theta}}^{\circ S}(\boldsymbol{y}) &= \boldsymbol{g}_{\boldsymbol{\theta}}^{\circ S}(\boldsymbol{z}_0) = \boldsymbol{z}_S \approx \boldsymbol{z}(\delta S) \\ &= \boldsymbol{z}(T) = \boldsymbol{\Phi}(\boldsymbol{z}_0, \boldsymbol{\theta}) = \boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{\theta}). \end{aligned}$$

Our results imply that $\mathcal{L}_2 \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{\circ S} \circ \boldsymbol{\mathcal{L}}_1$ has immense approximation power even if $\boldsymbol{g}_{\boldsymbol{\theta}}$ is realized by a fixed-size ReLU network, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are affine maps matching the dimensions. Define $\boldsymbol{F} : \mathbb{R}^{n+1} \times \boldsymbol{\Theta} \to \mathbb{R}^n$ via

$$\boldsymbol{F}(\boldsymbol{y}, t, \boldsymbol{\theta}) \coloneqq \big(\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{y}\big)/\delta, \tag{2}$$

where $\boldsymbol{g}_{\boldsymbol{\theta}}$ is realized by a fixed-size ReLU network. Then, the function $\mathcal{L}_2 \circ \boldsymbol{\Phi}(\cdot, \boldsymbol{\theta}) \circ \boldsymbol{\mathcal{L}}_1$, modelled by a continuous-depth network, can approximate $\mathcal{L}_2 \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{\circ S} \circ \boldsymbol{\mathcal{L}}_1$ well and hence also has immense approximation power. The definition of the dynamics function $\boldsymbol{F}$ in Equation (2) implies that $\boldsymbol{F}(\boldsymbol{y}, t, \boldsymbol{\theta})$ is independent of t for any $(\boldsymbol{y}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \boldsymbol{\Theta}$ and $\boldsymbol{F}$ can also be realized by a fixed-size ReLU network. In short, we have shown that a continuous-depth network can also have immense approximation power even if its dynamics function is time-independent and realized by a fixed-size ReLU network. One may refer to Section 2.1 for a further discussion on dynamical systems.

The remaining sections of this paper are structured as follows. In Section 2, we discuss the connections between our results and existing work. Section 3 outlines the main ideas behind the proofs of Theorems 1.1 and 1.3. Next, in Section 4, we provide two simple experiments to numerically validate our theoretical results. Finally, Section 5 concludes this paper with a brief discussion.

2. Related Work

In this section, we will provide a comprehensive overview of previous research that is pertinent to our results. We commence by emphasizing the correlation between deep learning and dynamical systems. Subsequently, we delve into the subject of parameter-sharing schemes in neural networks. Finally, we compare our results with existing research from the standpoint of function approximation.

2.1. Deep Learning via Dynamical Systems

A dynamical system is a mathematical framework that describes the evolution of a system over time. Its origins can be traced back to Newtonian mechanics. For a comprehensive overview of the history of dynamical systems, one may refer to (Holmes, 2007). In general, a dynamical system consists of two fundamental components. First, we have the state variable(s), which represent the variables that fully describe the state of the system. These variables capture the relevant properties or quantities of interest in the system. The second component is the time evolution rule, which specifies how the future states of the system evolve from the current state. It provides the mathematical equations or rules that govern the dynamics of the system over time. By studying the behavior and properties of dynamical systems, we gain insights into how systems change and develop over time. This framework has found applications in various fields, including physics, biology, economics, and computer science. In the context of deep learning, the connection to dynamical systems highlights the temporal aspect of learning and the potential for capturing complex dynamics in neural networks.

In recent years, there has been a growing body of research establishing connections between dynamical systems and deep learning. One such work (E, 2017) introduces a novel concept that interprets the discretization of a continuous dynamical system as a continuous-depth residual network. This approach utilizes continuous dynamical systems to model high-dimensional nonlinear functions commonly encountered in machine learning tasks. Another notable contribution by the authors in (Chen et al., 2018) parameterizes the derivative of the hidden state using a neural network, introducing continuous-depth residual networks. This work highlights several advantages of continuous-depth models, including constant memory cost. The study in (Li et al., 2023) establishes general sufficient conditions for the universal approximation property of continuous-depth residual networks, further connecting the dynamical systems approach to deep learning. A similar result is demonstrated in (Li et al., 2022), where the focus shifts to specific invariant functions instead of generic continuous functions. Additionally, the universal approximation property of deep fully convolutional networks is explored from

the perspective of dynamical systems in (Lin et al., 2022). The authors demonstrate that deep residual fully convolutional networks, along with their continuous-depth counterparts of constant channel width, can achieve the universal approximation of specific symmetric functions. These studies serve as exemplary demonstrations of the endeavors made to establish connections between dynamical systems and deep learning. They explore the potential benefits and theoretical foundations of continuous-depth models in various contexts.

2.2. Parameter Sharing in Neural Networks

In recent years, deep neural network models have demonstrated notable accomplishments across various domains. Nonetheless, the growing size of deep neural network models frequently introduces complexities in terms of computation and memory usage. To tackle these challenges, several techniques for model compression and acceleration have been developed, many of which involve the concept of parameter sharing. Parameter-sharing schemes are utilized in neural networks to minimize the total number of parameters, resulting in reduced memory and computational requirements. Our network architecture, which involves the repeated compositions of a single fixed-size network, can be viewed as a particular instance of a parameter-sharing scheme in neural networks.

To the best of our knowledge, parameter-sharing schemes in neural networks can be broadly categorized into three basic cases. The first case involves sharing parameters within the same layer, as seen in convolutional neural networks (CNNs), where kernels (filters) are shared across all image positions. The second case entails sharing parameters among different layers of neural networks, as in recurrent neural networks (RNNs). Our network architecture follows this second scheme by sharing parameters through repeated compositions of a single fixed-size network. We demonstrate that this approach can yield immense approximation capabilities by repeating a fixed number of parameters. In addition to these two parameter-sharing schemes, there is also the practice of sharing parameters across different neural networks or models, which is often employed in multi-task learning scenarios. For further insights into parameter sharing in neural networks, interested readers can refer to the references (Savarese & Maire, 2019; Wang et al., 2020; Plummer et al., 2022; Wang et al., 2020; Zhang et al., 2022; Wallingford et al., 2022).

2.3. Discussion from an Approximation Perspective

The approximation power of neural networks has been extensively studied, with numerous publications focusing on constructing various neural networks to approximate a wide range of target functions. Some notable examples include (Cybenko, 1989; Hornik et al., 1989; Barron, 1993; Yarotsky, 2018; 2017; Bölcskei et al., 2019; Zhou, 2020; Chui et al., 2018; Gribonval et al., 2022; Gühring et al., 2020; Suzuki, 2019; Nakada & Imaizumi, 2020; Chen et al., 2019; Bao et al., 2023; Li et al., 2023; Montanelli & Yang, 2020; Shen et al., 2019; 2020; Lu et al., 2021; Zhang, 2020; Shen et al., 2022b;a). In the early stages of this field, the focus was on exploring the universal approximation power of one-hidden-layer networks. The universal approximation theorem (Cybenko, 1989; Hornik, 1991; Hornik et al., 1989) demonstrated that a sufficiently large neural network can approximate a certain type of target function arbitrarily well, without explicitly estimating the approximation error in terms of the network size. Subsequent work, such as (Barron, 1993; Barron & Klusowski, 2018), analyzed the approximation error of one-hidden-layer networks of width n and showed an asymptotic approximation error of $\mathcal{O}(n^{-1/2})$ in the L^2-norm for target functions with certain smoothness properties.

Recent research has placed significant emphasis on the approximation of deep neural networks. Notably, the findings presented in (Shen et al., 2020; Yarotsky, 2018; Zhang, 2020) indicate that ReLU networks with n parameters can achieve an optimal approximation error of $\mathcal{O}(n^{-2/d})$ when approximating 1-Lipschitz continuous functions on $[0,1]^d$. However, it is crucial to recognize that this optimal approximation rate suffers from the curse of dimensionality. To overcome the limitations imposed by the curse of dimensionality and achieve better approximation errors, various approaches have been proposed and explored. These approaches aim to enhance the quality of approximation or even directly address the challenges arising from the curse of dimensionality. One approach is to consider smaller function spaces, such as smooth functions (Lu et al., 2021; Yarotsky & Zhevnerchuk, 2020), band-limited functions (Montanelli et al., 2021), and Barron spaces (Barron & Klusowski, 2018; Barron, 1993; E et al., 2022). By restricting the class of functions being approximated, it is possible to achieve better approximation errors with neural networks. Another approach is to design new network architectures that can improve the approximation capabilities. Examples of such architectures include Floor-ReLU networks (Shen et al., 2021a), Floor-Exponential-Step networks (Shen et al., 2021b), (Sin, ReLU, 2^x)-activated networks (Jiao et al., 2021), and three-dimensional networks (Shen et al., 2022). By exploring different function spaces and designing novel network architectures, researchers have been able to push the limits of approximation accuracy in neural networks, providing more flexibility and better performance for various tasks. It is important to note that the literature on the approximation analysis of deep neural networks is vast, and the publications mentioned here represent only a subset of the existing research.

Many other approaches, techniques, and architectures have been proposed to address the challenge of improving the approximation error for specific function classes.

In this paper, we propose a specific neural network architecture generated by repeated compositions of a single fixed-size network. Theorems 1.1 and 1.3 demonstrate that repeating a small ReLU network block can enhance the approximation power of our network architecture. We will conduct experiments in Section 4 to numerically verify our theoretical results and evaluate the approximation capabilities of our network architecture.

3. Ideas for Proving Theorems 1.1 and 1.3

Let us outline the main ideas behind the proofs of Theorems 1.1 and 1.3. During the proofs, our main approach involves constructing a piecewise constant function to approximate the desired continuous function. However, the continuity of ReLU networks poses a challenge in uniformly approximating piecewise constant functions. To bridge this gap, we first design ReLU networks to realize piecewise constant functions outside a sufficiently small region to approximate the target function well. Then, we will introduce a theorem to deal specifically with the approximation inside this small region for achieving uniform approximation.

Based on the aforementioned ideas, let us delve into the specific details. We divide $[0,1]^d$ into a union of "important" cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and a small region Ω, where K is a proper integer determined later. Each $Q_{\boldsymbol{\beta}}$ is associated with a representative $\boldsymbol{x}_{\boldsymbol{\beta}} \in Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. See Figure 1 for an illustration of $\boldsymbol{x}_{\boldsymbol{\beta}}$, Ω, and $Q_{\boldsymbol{\beta}}$. Then, the construction of the desired network approximating the target function can be divided into three steps as follows.

1. First, we design a sub-network to realize a vector-valued function $\boldsymbol{\Phi}_1$ mapping the whole cube $Q_{\boldsymbol{\beta}}$ to its index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$. That is, $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

2. Next, we design a sub-network to realize a function ϕ_2 mapping $\boldsymbol{\beta}$ approximately to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$. That is, $\phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

3. Finally, by defining $\phi \coloneqq \phi_2 \circ \boldsymbol{\Phi}_1$, we have $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Additionally, we must also address the approximation occurring within Ω and demonstrate that $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$ can be represented in the desired form $\phi = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are affine linear maps and $\boldsymbol{g}$ is realized by a fixed-size ReLU network.

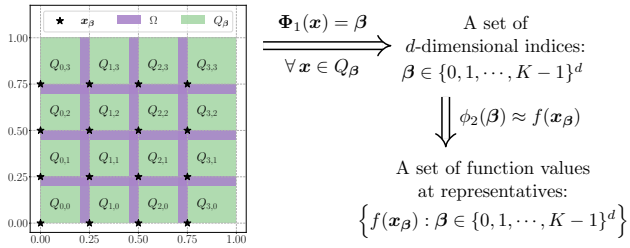


Figure 1. An illustration of the ideas for constructing the desired function $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$. Note that $\phi \approx f$ outside Ω since $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

See Figure 1 for an illustration of these three steps. More details on these three steps can be found below.

Step 1: Constructing $\boldsymbol{\Phi}_1$.

As mentioned previously, the aim of $\boldsymbol{\Phi}_1$ is to map $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Note that $\boldsymbol{\Phi}_1$ can be defined/constructed via

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big(\phi_1(x_1),\, \cdots,\, \phi_1(x_d)\big)$$

for any $\boldsymbol{x} = (x_1,\cdots,x_d) \in \mathbb{R}^d$, where $\phi_1 : \mathbb{R} \to \mathbb{R}$ is a step function outside a small region and hence can be realized by a ReLU network. It is generally challenging to design a ReLU network with a limited budget and the required architecture to realize such a function ϕ_1. Thus, we establish a proposition, Proposition 3.1 below, to do this step and place its proof in Section C of the appendix.

Proposition 3.1. *Given any $\delta \in (0,1)$ and $n, m \in \mathbb{N}^+$ with $n \le m$, there exist $\boldsymbol{g} \in \mathcal{NN}\{9,\, 1;\, \mathbb{R}^5 \to \mathbb{R}^5\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R} \to \mathbb{R}^5$ and $\mathcal{L}_2 : \mathbb{R}^5 \to \mathbb{R}$ such that*

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \boldsymbol{\mathcal{L}}_1(x) = k$$

if $x \in \big[k,\, k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}\big]$ for $k = 0,1,\cdots,n-1$.

Step 2: Constructing ϕ_2.

The objective of ϕ_2 is to map $\boldsymbol{\beta}$ approximately to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. It is important to note that, during in the construction of ϕ_2, we only need to care about the values of ϕ_2 sampled inside the set $\{0,1,\cdots,K-1\}^d$, which is a key point to ease the design of a ReLU network realizing ϕ_2. Indeed, if we can define a proper affine linear map $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$, then we only need to construct $\widetilde{\phi}_2 : \mathbb{R} \to \mathbb{R}$ to map $\mathcal{L}(\boldsymbol{\beta})$ approximatly to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$ since $\phi_2 = \widetilde{\phi}_2 \circ \mathcal{L}$ can map $\boldsymbol{\beta}$ approximately to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$. It is still challenging to construct a ReLU network with a limited budget and the required architecture to realize $\widetilde{\phi}_2$. Thus, we establish Proposition 3.2 below to simplify the construction of $\widetilde{\phi}_2$. The proof of Proposition 3.2 is complicated and hence is placed in Section D of the appendix.

Proposition 3.2. *Given any $\varepsilon > 0$, $n, m \in \mathbb{N}^+$ with $n \leq m$, and $y_k \geq 0$ for $k = 0, 1, \cdots, n-1$ with*

$$|y_k - y_{k-1}| \leq \varepsilon \quad \text{for } k = 1, 2, \cdots, n-1,$$

there exist $\boldsymbol{g} \in \mathcal{NN}\{16,\, 2;\, \mathbb{R}^6 \to \mathbb{R}^6\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R} \to \mathbb{R}^6$ and $\mathcal{L}_2 : \mathbb{R}^6 \to \mathbb{R}$ such that

$$\left|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \boldsymbol{\mathcal{L}}_1(k) - y_k\right| \leq \varepsilon \quad \text{for } k = 0, 1, \cdots, n-1.$$

Step 3: Representing $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$ properly.

With $\boldsymbol{\Phi}_1$ and ϕ_2 constructed in the first two steps, we can define $\phi \coloneqq \phi_2 \circ \boldsymbol{\Phi}_1$ and we have

$$\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$$

for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. That means ϕ can approximate f well outside Ω. By making ϕ bounded and Ω sufficiently small, we can easily control the L^p-norm approximation error to prove Theorem 1.1 for any $p \in [1, \infty)$. To prove Theorem 1.3, we require ϕ to pointwise approximate f well. To this end, we use the idea of Lemma 3.11 in (Zhang, 2020) (or Lemma 3.4 in (Lu et al., 2021)) to control the approximation error inside a small region.

Apart from a good approximation error, we also need to show that ϕ can be represented as the desired form $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are two affine linear maps and $\boldsymbol{g}$ is realized by a fixed-size ReLU network. Note that $\boldsymbol{\Phi}_1$ and ϕ_2 are constructed based on Propositions 3.1 and 3.2, respectively. Thus, both $\boldsymbol{\Phi}_1$ and ϕ_2 are expected to have the following form:

$$\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1,$$

where $\boldsymbol{\mathcal{L}}_1, \boldsymbol{\mathcal{L}}_2$ are affine linear maps and $\boldsymbol{g}$ is realized by fixed-size ReLU networks. Then, $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$ are expected to have the following form:

$$\phi = \widetilde{\mathcal{L}}_3 \circ \boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1, \tag{3}$$

where $\widetilde{\boldsymbol{\mathcal{L}}}_1, \widetilde{\boldsymbol{\mathcal{L}}}_2, \widetilde{\mathcal{L}}_3$ are affine linear maps and $\boldsymbol{g}_1, \boldsymbol{g}_2$ are realized by fixed-size ReLU networks. It is not trivial to convert the form in Equation (3) to the desired form. A proposition is established to facilitate such a conversion.

Proposition 3.3. *Let $\widetilde{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$, $\widetilde{\boldsymbol{\mathcal{L}}}_2 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, and $\widetilde{\boldsymbol{\mathcal{L}}}_3 : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$ be three affine linear maps. Suppose*

$$\boldsymbol{g}_i \in \mathcal{NN}\{N_i,\, L_i;\, \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}\}$$

and $r_i \in \mathbb{N}^+$ for $i = 1, 2$. For any $A > 0$ and $d \in \mathbb{N}^+$ with $d \geq \max\{d_1, d_2\}$, there exist $\boldsymbol{g} \in \mathcal{NN}\{N_1 + N_2 + 6d + 2,\, \max\{L_1 + 2, L_2 + 1\};\, \mathbb{R}^{d+2} \to \mathbb{R}^{d+2}\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d+2}$ and $\boldsymbol{\mathcal{L}}_2 : \mathbb{R}^{d+2} \to \mathbb{R}^{d_3}$ such that

$$\widetilde{\boldsymbol{\mathcal{L}}}_3 \circ \boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = \boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}^{\circ(r_1+r_2+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x})$$

for any $\boldsymbol{x} \in [-A, A]^{d_0}$.

The proof of Proposition 3.3 is technical and hence is deferred to Section E of the appendix.

4. Numerical Experiments

The primary objective of this section is to numerically validate the theoretical results presented in Theorems 1.1 and 1.3. To achieve this goal, we have designed two distinct experiments. The first experiment, described in Section 4.1, focuses on a function approximation task. Our aim is to demonstrate that increasing r in our RCNet architecture, denoted as $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$, improves the error of function approximation. In this architecture, $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ represent affine linear maps, while $\boldsymbol{g}$ represents a ReLU network block. The second experiment, outlined in Section 4.2, focuses on a classification task. We intend to illustrate that increasing the value of r in our RCNet architecture $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ leads to enhanced classification performance. By evaluating the accuracy of the classification results, we can empirically validate the advantages of incorporating multiple compositions of the fixed-size ReLU network block. Through these experiments, our goal is to provide numerical evidence that supports the theoretical claims and showcases the potential of our RCNet architecture in terms of improving approximation power. The results obtained from these experiments will contribute to a comprehensive understanding of the practical implications and advantages of our network design.

Next, let us briefly discuss the expected impact of increasing r in our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ on the experiment results. In this discussion, we will focus on the ReLU activation function and consider the three main sources of errors in the test results: approximation error, generalization error, and optimization error. In our experiments where r is small, we assume that the optimization error is well-controlled due to the utilization of a good optimization algorithm. Thus, we can primarily focus on the effects of increasing r on the approximation and generalization errors. Increasing r has the potential to reduce the approximation error, as demonstrated by our theoretical results. However, it may also lead to an increase in the generalization error. Therefore, the performance improvement associated with larger values of r depends on whether the approximation error or the generalization error dominates. If the approximation error is the leading term, then increasing r can be beneficial. To emphasize the approximation error, we design our first experiment to involve a sufficiently complex target function. By choosing a challenging binary classification problem for our second experiment, we create conditions where the approximation error is relatively large. In both experiments, a sufficient number of samples are generated to control the generalization error, ensuring that it does not overshadow the effects of the approximation error. By

carefully setting up these experiments and controlling the different sources of errors, we can gain insights into the impact of increasing r in our RCNet architecture.

4.1. Function Approximation

To evaluate and compare the approximation capabilities of our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ across various values of r and n, we will utilize it for a function approximation task. This architecture comprises two affine linear maps, $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$, along with a ReLU network block $\boldsymbol{g}_n$. To facilitate a comprehensive comparison, we have specifically chosen a target function f that exhibits a high degree of complexity. The function $f : [0,1]^2 \to \mathbb{R}$ is defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{2}\sum_{j=1}^{2} a_{i,j} \sin(b_i x_i + c_{i,j} x_i x_j) \cos(b_j x_j + d_{i,j} x_i^2)$$

for any $\boldsymbol{x} = (x_1, x_2) \in [0,1]^2$, where the coefficient matrices are given by

$$(a_{i,j}) = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}, \quad (b_i) = \begin{bmatrix} 2\pi \\ 4\pi \end{bmatrix},$$

$$(c_{i,j}) = \begin{bmatrix} 2\pi & 4\pi \\ 8\pi & 4\pi \end{bmatrix}, \quad \text{and} \quad (d_{i,j}) = \begin{bmatrix} 4\pi & 6\pi \\ 8\pi & 6\pi \end{bmatrix}.$$

To visually represent the target function f, we have included illustrations in Figure 2. By choosing this intricate function as our target, we can effectively assess the approximation capability of our network architecture across various values of r and n.

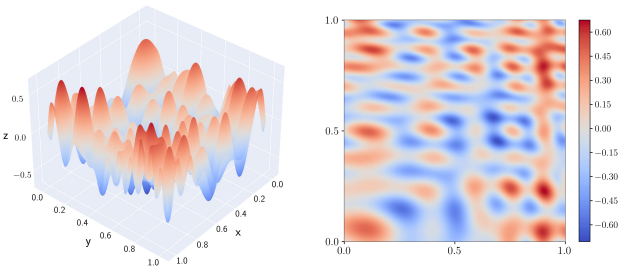


Figure 2. Illustrations of the target function f.

In this experiment, we will employ the RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ to approximate the target function f for different values of r and n. Specifically, we consider $r = 1, 2, 3, 4$ and $n = 100, 200$. The ReLU network block $\boldsymbol{g}_n$ is constructed by combining an affine linear map and the ReLU activation function, i.e., $\boldsymbol{g}_n$ is defined as

$$\boldsymbol{g}_n(\boldsymbol{x}) \coloneqq \sigma(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$$

for any $\boldsymbol{x} \in \mathbb{R}^n$, where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are parameters and σ is the ReLU activation function that can be applied element-wise to a vector. Then, the dimensions of the input and output for the two affine linear maps, $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^2 \to \mathbb{R}^n$ and $\mathcal{L}_2 : \mathbb{R}^n \to \mathbb{R}$, are determined accordingly. Notably, the ReLU network block $\boldsymbol{g}_n$ consists of $n^2 + n$ parameters. The affine linear map $\boldsymbol{\mathcal{L}}_1$ contains $2n + n = 3n$ parameters, while $\mathcal{L}_2$ has $n + 1$ parameters. Consequently, the total number of parameters in $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ is $(n^2+n)+3n+(n+1) = n^2+5n+1$ for different values of r and n. It is essential to note that when $r \geq 2$, the parameters of $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ are partially shared through repetitions of the ReLU network block $\boldsymbol{g}_n$. Our objective is to provide numerical evidence demonstrating that increasing the value of r results in improved test losses for each fixed n.

Before presenting the numerical results, let us delve into the hyperparameters utilized for training our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for varying values of r and n, specifically $r = 1, 2, 3, 4$ and $n = 100, 200$. To generate the training and test samples, we employ the uniform distribution, resulting in 10^6 training samples and 10^5 test samples in $[0,1]^2$. During the training process, we utilize the RAdam optimization method (Liu et al., 2020), which aids in optimizing the network parameters. We set the mini-batch size for training to 500, which signifies the number of training samples processed in each iteration. Our training process comprises a total of 500 epochs, representing complete passes through the training dataset. The learning rate is adjusted every 5 epochs. More specifically, the learning rate during epochs $5(i-1)+1$ to $5i$ is set to $0.002 \times 0.9^{i-1}$ for $i = 1, 2, \cdots, 100$. This adaptive adjustment allows for fine-tuning the model during training. To train our model, we employ the mean squared error (MSE) loss function, which measures the average squared difference between the network-generated function and the target function. To ensure the reliability of our experiment, we repeat it 12 times. From these repetitions, we discard 3 top-performing and 3 bottom-performing trials based on the average test accuracy of the last 100 epochs. The target accuracy is then determined by taking the average of the test accuracies from the remaining 6 trials for each epoch.

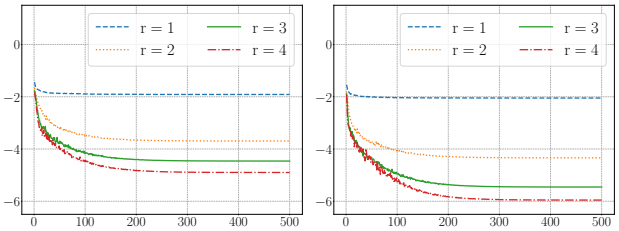
We are now prepared to present the experiment results that compare the numerical performances of $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for different values of r and n, specifically $r = 1, 2, 3, 4$ and $n = 100, 200$. The test losses over the last 100 epochs are averaged to obtain the target losses, considering two types of loss functions: mean squared error (MSE) and maximum (MAX) loss functions. Table 1 provides a comprehensive comparison of the numerical results obtained from $\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for the given values of r and n. Additionally, Figure 3 illustrates the test losses measured in MSE on a logarithmic scale, allowing for an intuitive comparison. The values presented in Table 1 and the trends observed in Figure 3 clearly indicate a significant improvement in test losses with increasing values of r. These findings align with the theoretical results stated in Theorems 1.1 and 1.3,

providing further confirmation of the effectiveness of increasing r.

Lastly, it is crucial to acknowledge that further increasing the value of r may not lead to additional improvements in the results due to the inherent challenges involved in optimizing deep learning models. Issues such as local minima, saddle points, and vanishing gradients make it increasingly difficult to identify the global minimizer, particularly for larger values of r.

Table 1. Test loss comparison.

$\mathcal{L}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \mathcal{L}_1$	$n = 100$		$n = 200$	
	MSE	MAX	MSE	MAX
$r = 1$	1.22×10^{-2}	4.21×10^{-1}	8.92×10^{-3}	4.05×10^{-1}
$r = 2$	2.02×10^{-4}	1.17×10^{-1}	4.60×10^{-5}	7.28×10^{-2}
$r = 3$	3.46×10^{-5}	4.65×10^{-2}	3.51×10^{-6}	1.66×10^{-2}
$r = 4$	1.27×10^{-5}	2.77×10^{-2}	1.11×10^{-6}	8.61×10^{-3}



(a) $n = 100$. (b) $n = 200$.

Figure 3. Test losses measured in MSE across epochs: the x-axis represents the epoch number, while the y-axis corresponds to the base-10 logarithm of the test loss.

4.2. Classification

To assess and compare the approximation capabilities of our RCNet architecture $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ across different values of r and n, we will employ it for a classification task. This architecture consists of two affine linear maps, $\boldsymbol{\mathcal{L}}_1$ and $\boldsymbol{\mathcal{L}}_2$, along with a ReLU network block $\boldsymbol{g}_n$. For a comprehensive comparison, we have selected a complex binary classification experiment utilizing the Archimedean spiral, as proposed in (Shen et al., 2022). The objective of this classification problem is to accurately classify samples from two distinct sets, denoted as $\mathcal{S}_0$ and $\mathcal{S}_1$. These two sets are constructed based on the Archimedean spiral, as illustrated in Figure 4.

Let us delve into the construction details of the sets $\mathcal{S}_0$ and $\mathcal{S}_1$. An Archimedean spiral can be represented by the equation $r = a + b\theta$ in polar coordinates (r, θ) for proper $a, b \in \mathbb{R}$. We begin by defining two curves

$$\widetilde{\mathcal{C}}_i \coloneqq \Big\{ \big(r_i \cos\theta,\ r_i \sin\theta\big) : r_i = a_i + b_i\theta,\ \ \theta \in [0, s\pi] \Big\}$$

for $i = 0, 1$, where $a_0 = 0$, $a_1 = 1$, $b_0 = b_1 = 1/\pi$, and

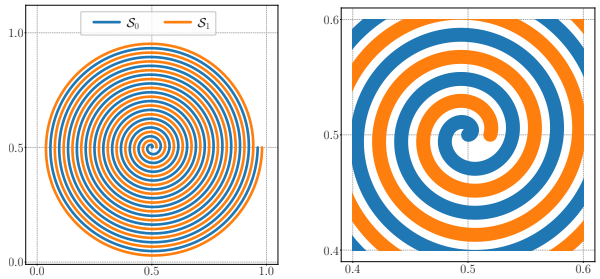


Figure 4. Illustrations for $\mathcal{S}_0$ and $\mathcal{S}_1$.

$s = 24$. Next, we normalize $\widetilde{\mathcal{C}}_i$ to obtain $\mathcal{C}_i \subseteq [0, 1]^2$ for each $i \in \{0, 1\}$, where $\mathcal{C}_i$ is defined as

$$\mathcal{C}_i \coloneqq \left\{ (x, y) : x = \tfrac{\widetilde{x}+(s+2)}{2(s+2)},\ y = \tfrac{\widetilde{y}+(s+2)}{2(s+2)},\ (\widetilde{x}, \widetilde{y}) \in \widetilde{\mathcal{C}}_i \right\}$$

for $i = 0, 1$. With $\mathcal{C}_0$ and $\mathcal{C}_1$ defined, we can construct the target sets $\mathcal{S}_0$ and $\mathcal{S}_1$ as

$$\mathcal{S}_i \coloneqq \left\{ (x, y) : \sqrt{(x-u)^2 + (y-v)^2} \le \varepsilon,\ (u, v) \in \mathcal{C}_i \right\}$$

for $i = 0, 1$, where $\varepsilon = 0.006$ in our experiments. Refer to Figure 4 for illustrations of $\mathcal{S}_0$ and $\mathcal{S}_1$.

In this experiment, we will employ the network architecture $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ to classify samples in $\mathcal{S}_0 \cup \mathcal{S}_1$ for different values of r and n, specifically $r = 1, 2, 3, 4$ and $n = 30, 40$. Here, $\boldsymbol{\mathcal{L}}_1$ and $\boldsymbol{\mathcal{L}}_2$ represent two affine linear maps, and $\boldsymbol{g}_n$ corresponds to a ReLU network block. The construction of the ReLU network block $\boldsymbol{g}_n$ involves combining an affine linear map with the ReLU activation function. Mathematically, $\boldsymbol{g}_n$ is defined as

$$\boldsymbol{g}_n(\boldsymbol{x}) \coloneqq \sigma(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$$

for any $\boldsymbol{x} \in \mathbb{R}^n$, where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are parameters and σ denotes the ReLU activation function that can be applied element-wise to a vector. Subsequently, the input and output dimensions of the two affine linear maps, $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^2 \to \mathbb{R}^n$ and $\boldsymbol{\mathcal{L}}_2 : \mathbb{R}^n \to \mathbb{R}^2$, are appropriately determined. The total number of parameters in $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ can be easily verified to be $(2n+2)+(n^2+n)+(2n+n) = n^2+6n+2$ for different values of r and n. An important observation is that for $r \ge 2$, the parameters in $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ are partially shared due to the repeated utilization of the ReLU network block $\boldsymbol{g}_n$. Our objective is to provide numerical evidence demonstrating that increasing the value of r results in improved test accuracies for each fixed n.

Before proceeding with the numerical results, let us provide an overview of the hyperparameters employed in training our network architecture $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for different values of r and n, specifically $r = 1, 2, 3, 4$ and $n = 30, 40$. First, we generate training and test samples from $\mathcal{S}_0$ and $\mathcal{S}_1$

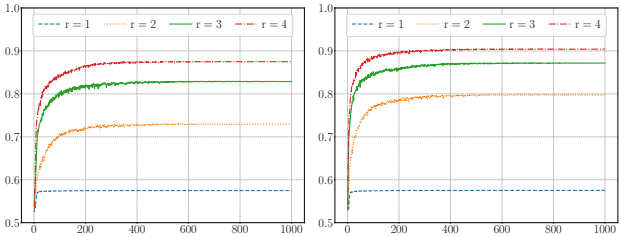
using the uniform distribution. Specifically, we randomly generate 3×10^5 training samples and 3×10^4 test samples for each class. These 6×10^5 training samples are used for network training, while 6×10^4 test samples are utilized to compute the test accuracy. For optimization, we employ the RAdam method (Liu et al., 2020). The training process consists of 1000 epochs with a mini-batch size of 300. The learning rate is defined as $0.001 \times 0.95^{i-1}$ during epochs $5(i-1)+1$ to $5i$ for $i = 1, 2, \cdots, 200$. To evaluate the model output, we apply the softmax activation function to the network output and employ the cross-entropy loss function to measure the loss between the target function and the network output. To guarantee the standardization of training and test samples, we perform a rescaling procedure to adjust their mean to 0 and standard deviation to 1. To ensure reliability of our results, we conduct the experiment 12 times. Among these trials, we exclude 3 top-performing and 3 bottom-performing trials based on the average test accuracy over the last 100 epochs. The target accuracy is then determined by averaging the test accuracies from the remaining 6 trials for each epoch.

Let us now present the results of our experiments by comparing the numerical performances of $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for for varying values of r and n, specifically $r = 1, 2, 3, 4$ and $n = 30, 40$. The target test accuracy is computed by averaging the test accuracies over the last 100 epochs. Table 2 provides a comprehensive overview of the test accuracies obtained by $\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ for different values of r and n. To enhance the information presented in Table 2, Figure 5 serves as a complementary visual representation. It showcases graphical depictions of the data, enabling an intuitive comparison and facilitating a visual analysis of the performance trends. The results presented in Table 2 combined with the trends observed in Figure 5 confirm our initial expectation that increasing the value of r leads to improved test accuracies. These experiment results provide compelling numerical evidence demonstrating the effectiveness of increasing r, which aligns with the theoretical results stated in Theorems 1.1 and 1.3.

Finally, it is important to note that further increasing r may not necessarily result in additional improvements in the results. This is because optimizing deep learning models is notoriously challenging, as it involves various issues such as local minima, saddle points, and vanishing gradients. In our experiments, the primary difficulty lies in identifying the global minimizer, especially when dealing with large values of r.

Table 2. Test accuracy comparison.

$\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}_n^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$	$r = 1$	$r = 2$	$r = 3$	$r = 4$
$n = 30$	0.574781	0.729258	0.828750	0.874866
$n = 40$	0.575149	0.797345	0.871617	0.904041



(a) $n = 30$. (b) $n = 40$.

Figure 5. Test accuracies across epochs: the x-axis represents the epoch number, while the y-axis corresponds to the test accuracy.

5. Conclusion

This paper investigates the expressive power of deep neural networks from the perspective of function compositions. We demonstrate that the repeated compositions of a single fixed-size ReLU network exhibit surprising expressive power, despite the limited expressive capabilities of the individual network itself. As shown in Theorems 1.1 and 1.3, our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ can approximate any continuous function $f \in C([0,1]^d)$ with an error $\mathcal{O}\big(\omega_f(r^{-1/d})\big)$. Here, $\boldsymbol{g}$ represents a fixed-size ReLU network, while $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ correspond to two affine linear maps matching the dimensions. Furthermore, we explore the connection between our findings and dynamical systems. Our results reveal that a continuous-depth network generated through a dynamical system possesses enormous approximation capabilities, even when the dynamics function is time-independent and realized by a fixed-size ReLU network. Finally, we conduct experiments to provide numerical evidence that validates the theoretical results stated in Theorems 1.1 and 1.3.

It is worth mentioning that our analysis is currently focused on the ReLU activation function and fully connected network architectures. Extending our results to other activation functions, such as the sigmoid and tanh functions, as well as different neural network architectures, such as convolutional neural networks, would be of great interest for future research. Additionally, the numerical examples presented in this paper are relatively simple. Further exploration of the numerical performance of our network architecture and its application to real-world problems would be an intriguing direction for future studies.

Acknowledgements

We extend our gratitude to Zuowei Shen, Yimin Zhong, and Haomin Zhou for their valuable insights and constructive feedback on our theoretical proofs and numerical experiments. Their contributions greatly enhanced the quality and rigor of our work. Jianfeng Lu was partially supported by NSF grants CCF-1910571 and DMS-2012286. Hongkai Zhao was partially supported by NSF grant DMS-2012860.

References

Bao, C., Li, Q., Shen, Z., Tai, C., Wu, L., and Xiang, X. Approximation analysis of convolutional neural networks. *East Asian Journal on Applied Mathematics*, 13(3):524–549, 2023. ISSN 2079–7370. doi: https://doi.org/10.4208/eajam.2022-270.070123. URL http://global-sci.org/intro/article_detail/eajam/21721.html.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993. ISSN 0018-9448. URL https://doi.org/10.1109/18.256500.

Barron, A. R. and Klusowski, J. M. Approximation and estimation for high-dimensional deep learning networks. *arXiv e-prints*, art. arXiv:1809.03090, September 2018. URL https://arxiv.org/abs/1809.03090.

Bartlett, P., Maiorov, V., and Meir, R. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10(8):2159–2173, 1998. URL https://doi.org/10.1162/089976698300017016.

Bölcskei, H., Grohs, P., Kutyniok, G., and Petersen, P. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1):8–45, 2019. doi: 10.1137/18M118709X. URL https://doi.org/10.1137/18M118709X.

Chen, M., Jiang, H., Liao, W., and Zhao, T. Efficient approximation of deep ReLU networks for functions on low dimensional manifolds. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/fd95ec8df5dbeea25aa8e6c808bad583-Paper.pdf.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Chui, C. K., Lin, S.-B., and Zhou, D.-X. Construction of neural networks for realization of localized deep learning. *Frontiers in Applied Mathematics and Statistics*, 4:14, 2018. ISSN 2297-4687. doi: 10.3389/fams.2018.00014. URL https://www.frontiersin.org/article/10.3389/fams.2018.00014.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989. URL https://doi.org/10.1007/BF02551274.

E, W. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 2017. URL https://doi.org/10.1007/s40304-017-0103-z.

E, W., Ma, C., and Wu, L. The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55:369–406, 2022. URL https://doi.org/10.1007/s00365-021-09549-y.

Gribonval, R., Kutyniok, G., Nielsen, M., and Voigtlaender, F. Approximation spaces of deep neural networks. *Constructive Approximation*, 55:259–367, 2022. URL https://doi.org/10.1007/s00365-021-09543-4.

Gühring, I., Kutyniok, G., and Petersen, P. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Analysis and Applications*, 18(05):803–859, 2020. doi: 10.1142/S0219530519410021. URL https://doi.org/10.1142/S0219530519410021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016. URL https://doi.org/10.1109/CVPR.2016.90.

Holmes, P. History of dynamical systems. *Scholarpedia*, 2(5):1843, 2007. URL https://doi.org/10.4249/scholarpedia.1843.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL http://www.sciencedirect.com/science/article/pii/0893608091900009T.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL http://www.sciencedirect.com/science/article/pii/0893608089900208.

Jiao, Y., Lai, Y., Lu, X., Wang, F., Zhijian Yang, J., and Yang, Y. Deep neural networks with ReLU-Sine-Exponential activations break curse of dimensionality on Hölder class. *arXiv e-prints*, art. arXiv:2103.00542, February 2021. URL https://arxiv.org/abs/2103.00542.

Li, Q., Lin, T., and Shen, Z. Deep neural network approximation of invariant functions through dynamical systems. *arXiv e-prints*, art. arXiv:2208.08707, August 2022. URL https://arxiv.org/abs/2208.08707.

Li, Q., Lin, T., and Shen, Z. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 25(5):1671–1709, 2023. URL https://doi.org/10.4171/JEMS/1221.

Lin, T., Shen, Z., and Li, Q. On the universal approximation property of deep fully convolutional neural networks. *arXiv e-prints*, art. arXiv:2211.14047, November 2022. URL https://arxiv.org/abs/2211.14047.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

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. URL https://doi.org/10.1137/20M134695X.

Montanelli, H. and Yang, H. Error bounds for deep ReLU networks using the Kolmogorov-Arnold superposition theorem. *Neural Networks*, 129:1–6, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.12.013. URL http://www.sciencedirect.com/science/article/pii/S0893608019304058.

Montanelli, H., Yang, H., and Du, Q. Deep ReLU networks overcome the curse of dimensionality for bandlimited functions. *Journal of Computational Mathematics*, 39(6):801–815, 2021. ISSN 1991-7139. doi: https://doi.org/10.4208/jcm.2007-m2019-0239. URL http://global-sci.org/intro/article_detail/jcm/19912.html.

Nakada, R. and Imaizumi, M. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174):1–38, 2020. URL http://jmlr.org/papers/v21/20-002.html.

Plummer, B. A., Dryden, N., Frost, J., Hoefler, T., and Saenko, K. Neural parameter allocation search. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=srtIXtySfT4.

Savarese, P. and Maire, M. Learning implicitly recurrent CNNs through parameter sharing. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJgYxn09Fm.

Shen, Z., Yang, H., and Zhang, S. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.07.011. URL http://www.sciencedirect.com/science/article/pii/S0893608019301996.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28(5):1768–1811, 2020. ISSN 1991-7120. URL https://doi.org/10.4208/cicp.OA-2020-0149.

Shen, Z., Yang, H., and Zhang, S. Deep network with approximation error being reciprocal of width to power of square root of depth. *Neural Computation*, 33(4):1005–1036, 03 2021a. ISSN 0899-7667. doi: 10.1162/neco_a_01364. URL https://doi.org/10.1162/neco_a_01364.

Shen, Z., Yang, H., and Zhang, S. Neural network approximation: Three hidden layers are enough. *Neural Networks*, 141:160–173, 2021b. ISSN 0893-6080. URL https://doi.org/10.1016/j.neunet.2021.04.011.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons. *Journal of Machine Learning Research*, 23(276):1–60, 2022a. URL http://jmlr.org/papers/v23/21-1404.html.

Shen, Z., Yang, H., and Zhang, S. Deep network approximation in terms of intrinsic parameters. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19909–19934. PMLR, 17–23 Jul 2022b. URL https://proceedings.mlr.press/v162/shen22g.html.

Shen, Z., Yang, H., and Zhang, S. Neural network architecture beyond width and depth. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing*

Systems, volume 35, pp. 5669–5681. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/257be12f31dfa7cc158dda99822c6fd1-Abstract-Conference.html.

Suzuki, T. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1ebTsActm.

Wallingford, M., Li, H., Achille, A., Ravichandran, A., Fowlkes, C., Bhotika, R., and Soatto, S. Task adaptive parameter sharing for multi-task learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7551–7560, 2022. URL https://doi.org/10.1109/CVPR52688.2022.00741.

Wang, J., Bai, H., Wu, J., Shi, X., Huang, J., King, I., Lyu, M., and Cheng, J. Revisiting parameter sharing for automatic neural channel number search. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5991–6002. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/42cd63cb189c30ed03e42ce2c069566c-Paper.pdf.

Wang, Z., Cheng, X., Sapiro, G., and Qiu, Q. ACDC: Weight sharing in atom-coefficient decomposed convolution. *arXiv e-prints*, art. arXiv:2009.02386, September 2020. URL https://arxiv.org/abs/2009.02386.

Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2017.07.002. URL http://www.sciencedirect.com/science/article/pii/S0893608017301545.

Yarotsky, D. Optimal approximation of continuous functions by very deep ReLU networks. In Bubeck, S., Perchet, V., and Rigollet, P. (eds.), *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 639–649. PMLR, 06–09 Jul 2018. URL http://proceedings.mlr.press/v75/yarotsky18a.html.

Yarotsky, D. and Zhevnerchuk, A. The phase diagram of approximation rates for deep neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13005–13015. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/979a3f14bae523dc5101c52120c535e9-Paper.pdf.

Zhang, L., Yang, Q., Liu, X., and Guan, H. Rethinking hard-parameter sharing in multi-domain learning. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 01–06, 2022. URL https://doi.org/10.1109/ICME52920.2022.9859706.

Zhang, S. Deep neural network approximation via function compositions. *PhD Thesis, National University of Singapore*, 2020. URL https://scholarbank.nus.edu.sg/handle/10635/186064.

Zhou, D.-X. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2019.06.004. URL http://www.sciencedirect.com/science/article/pii/S1063520318302045.

Contents

1 Introduction **1**

2 Related Work **3**

2.1 Deep Learning via Dynamical Systems . . . 3

2.2 Parameter Sharing in Neural Networks . . . 4

2.3 Discussion from an Approximation Perspective . . . 4

3 Ideas for Proving Theorems 1.1 and 1.3 **5**

4 Numerical Experiments **6**

4.1 Function Approximation . . . 7

4.2 Classification . . . 8

5 Conclusion **9**

A Proofs of Theorems 1.1 and 1.3 **14**

A.1 Notations . . . 14

A.2 Proof of Theorem 1.1 with Theorem A.1 . . . 15

A.3 Proof of Theorem 1.3 with Theorem A.1 . . . 17

B Proof of Theorem A.1 with Propositions **20**

C Proof of Proposition 3.1 **24**

C.1 Proof of Proposition 3.1 with Lemma C.1 . . . 24

C.2 Proof of Lemma C.1 . . . 25

D Proof of Proposition 3.2 **28**

D.1 Proof of Proposition 3.2 with Lemma D.1 . . . 28

D.2 Proof of Lemma D.1 . . . 29

E Proof of Proposition 3.3 **31**

E.1 Proof of Proposition 3.3 with Lemma E.1 . . . 31

E.2 Proof of Lemma E.1 . . . 33

A. Proofs of Theorems 1.1 and 1.3

As we shall see later in the proofs of Theorems 1.1 and 1.3, our main approach involves constructing a piecewise constant function to approximate the desired continuous function. However, the inherent continuity of ReLU networks hinders their ability to uniformly approximate piecewise constant functions effectively. To address this limitation, we introduce the concept of the trifling region $\Omega([0,1]^d, K, \delta)$ as defined in Equation (4). By utilizing ReLU networks, we can accurately represent piecewise constant functions outside the trifling region.

To streamline the proofs of Theorems 1.1 and 1.3, we introduce an auxiliary theorem, referred to as Theorem A.1 below, where we disregard the approximation within the trifling region $\Omega([0,1]^d, K, \delta)$.

Theorem A.1. *Given a continuous function $f \in C([0,1]^d)$, for any $r \in \mathbb{N}^+$, there exist $\boldsymbol{g} \in \mathcal{NN}\big\{39d+24,\ 3;\ \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\big\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ and $\mathcal{L}_2 : \mathbb{R}^{5d+3} \to \mathbb{R}$ such that*

$$\big|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) - f(\boldsymbol{x})\big| \le 5\sqrt{d}\,\omega_f(r^{-1/d}) \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

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

The proof of Theorem A.1 will be presented in Section B. Assuming the validity of Theorem A.1, we will provide the detailed proofs of Theorems 1.1 and 1.3 in Sections A.2 and A.3, respectively. To enhance clarity, Section A.1 offers a concise overview of the notations employed throughout this paper.

A.1. Notations

Below is a summary of the fundamental notations employed in this paper.

- The set difference of two sets A and B is denoted as $A \backslash B := \{x : x \in A,\ x \notin B\}$.
- The sets of natural numbers (including 0), integers, rational numbers, and real numbers are denoted as $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$, respectively. Set $\mathbb{N}^+ = \mathbb{N} \backslash \{0\}$.
- The indicator (characteristic) function of a set A is denoted as $\mathbb{1}_A$, which takes the value 1 on elements of A and 0 otherwise.
- The floor and ceiling functions of a real number x are denoted as $\lfloor x \rfloor = \max\{n : n \le x,\ n \in \mathbb{Z}\}$ and $\lceil x \rceil = \min\{n : n \ge x,\ n \in \mathbb{Z}\}$.
- Vectors and matrices are represented by bold lowercase and uppercase letters, respectively. For example, $\boldsymbol{a} = (a_1, \cdots, a_d) \in \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a real matrix of size $m \times n$, and $\boldsymbol{A}^T$ denotes the transpose of $\boldsymbol{A}$.
- Slicing notation is used for a vector $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$, where $[\boldsymbol{x}]_{[n:m]}$ denotes a slice of $\boldsymbol{x}$ from its n-th to the m-th entries and $[\boldsymbol{x}]_{[n]}$ denotes the n-th entry of $\boldsymbol{x}$ for any $n, m \in \{1, 2, \cdots, d\}$ with $n \le m$. For example, if $\boldsymbol{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$, then $[5\boldsymbol{x}]_{[2:3]} = (5x_2, 5x_3)$ and $[6\boldsymbol{x}+1]_{[3]} = 6x_3 + 1$.
- Given any $p \in [1, \infty]$, the p-norm (or ℓ^p-norm) of a vector $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$ is defined via

  $$\|\boldsymbol{x}\|_p = \|\boldsymbol{x}\|_{\ell^p} := \big(|x_1|^p + \cdots + |x_d|^p\big)^{1/p} \quad \text{if } p \in [1, \infty)$$

  and

  $$\|\boldsymbol{x}\|_\infty = \|\boldsymbol{x}\|_{\ell^\infty} := \max\big\{|x_i| : i = 1, 2, \cdots, d\big\}.$$
- By convention, $\sum_{j=n}^{m} a_j = 0$ if $n > m$, no matter what a_j is for each j.
- For any $\theta = \sum_{i=1}^{n} \theta_i 2^{-i} \in [0, 1)$, we use $\operatorname{bin} 0.\theta_1\theta_2\cdots\theta_L$ to denote the binary representation of θ, i.e., $\theta = \sum_{i=1}^{n} \theta_i 2^{-i} = \operatorname{bin} 0.\theta_1\theta_2\cdots\theta_n$.
- Given any $K \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{K})$, we define a trifling region $\Omega([0,1]^d, K, \delta)$ of $[0,1]^d$ via

  $$\Omega([0,1]^d, K, \delta) := \bigcup_{j=1}^{d} \left\{ \boldsymbol{x} = (x_1, \cdots, x_d) \in [0,1]^d : x_j \in \bigcup_{k=1}^{K-1} \left(\tfrac{k}{K} - \delta,\ \tfrac{k}{K} \right) \right\}. \tag{4}$$

  In the degenerate case $K = 1$, $\Omega([0,1]^d, K, \delta) = \emptyset$. Figure 6 presents two examples of trifling regions.

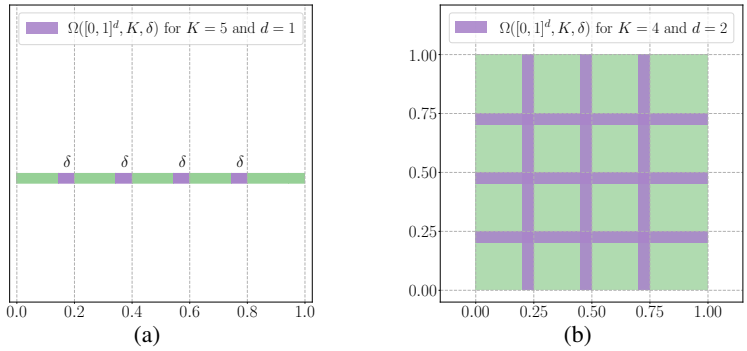


Figure 6. Two examples of trifling regions. (a) $K = 5, d = 1$. (b) $K = 4, d = 2$.

- The rectified linear unit (ReLU) is denoted as $\sigma(x) = \max\{0, x\}$ for any $x \in \mathbb{R}$. With a slight abuse of notation, we allow σ to be applied element-wise to a vector, i.e., $\sigma(\boldsymbol{x}) = \big(\sigma(x_1), \cdots, \sigma(x_d)\big)$ for any $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$.
- Suppose $\boldsymbol{\phi}$ is a function realized by a ReLU network, whether scalar or vector-valued. Then, $\boldsymbol{\phi}$ can be expressed as

$$\boldsymbol{x} = \widetilde{\boldsymbol{h}}_0 \xrightarrow[\boldsymbol{\mathcal{L}}_0]{\boldsymbol{W}_0,\, \boldsymbol{b}_0} \boldsymbol{h}_1 \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_1 \quad \cdots \quad \xrightarrow[\boldsymbol{\mathcal{L}}_{L-1}]{\boldsymbol{W}_{L-1},\, \boldsymbol{b}_{L-1}} \boldsymbol{h}_L \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_L \xrightarrow[\boldsymbol{\mathcal{L}}_L]{\boldsymbol{W}_L,\, \boldsymbol{b}_L} \boldsymbol{h}_{L+1} = \boldsymbol{\phi}(\boldsymbol{x}),$$

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

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

  and

$$\widetilde{\boldsymbol{h}}_i = \sigma(\boldsymbol{h}_i) \quad \text{for } i = 1, 2, \cdots, L.$$

  Furthermore, $\boldsymbol{\phi}$ can be expressed as a composition of functions. Specifically, it can be written as

$$\boldsymbol{\phi} = \boldsymbol{\mathcal{L}}_L \circ \sigma \circ \cdots \circ \boldsymbol{\mathcal{L}}_1 \circ \sigma \circ \boldsymbol{\mathcal{L}}_0.$$

  Refer to Figure 7 for an illustration.

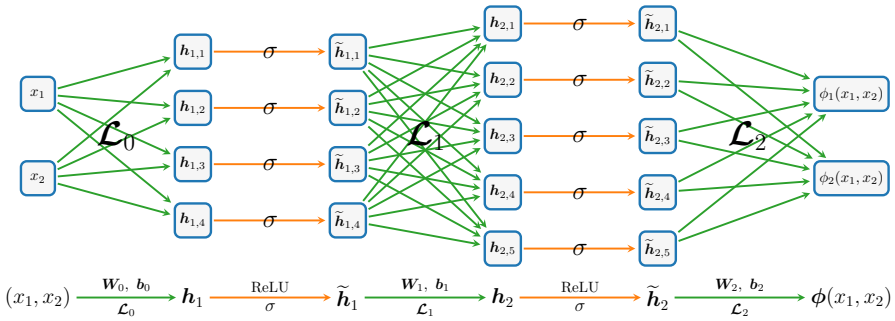


Figure 7. An example of a ReLU network of width 5 and depth 2. The network realizes a vector-valued function $\boldsymbol{\phi} = (\phi_1, \phi_2)$.

- A network is referred to as "a network of width N and depth L" if it satisfies the following conditions.
  - The number of neurons in each hidden layer of this network is less than or equal to N.
  - The number of hidden layers of this network is less than or equal to L.

A.2. Proof of Theorem 1.1 with Theorem A.1

By assuming the validity of Theorem A.1, we can proceed to prove Theorem 1.1.

Proof of Theorem 1.1. We assume that f is not a constant function, as considering constant functions would lead to a trivial case. Therefore, for any $t > 0$, we have $\omega_f(t) > 0$. Set $K = \lfloor r^{1/d} \rfloor$ and let $\delta \in (0, \frac{1}{3K}]$ be an arbitrary number determined later. By Theorem A.1, there exist

$$\boldsymbol{g}_1 \in \mathcal{NN}\big\{39d + 24,\ 3;\ \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\big\}$$

and two affine linear maps $\widehat{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ and $\widehat{\mathcal{L}}_2 : \mathbb{R}^{5d+3} \to \mathbb{R}$ such that

$$\big|\widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) - f(\boldsymbol{x})\big| \le 5\sqrt{d}\,\omega_f(r^{-1/d}) \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta). \tag{5}$$

That means the approximation error is well controlled outside the trifling region $\Omega([0,1]^d, K, \delta)$. To control the L^p-norm of $\widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1 - f$, we need to further bound it inside $\Omega([0,1]^d, K, \delta)$. To this end, we define

$$g_2(x) \coloneqq \begin{cases} M & \text{if } x \ge M \\ x & \text{if } |x| < M \\ -M & \text{if } x \le -M, \end{cases} \qquad \text{where } M = M_f = \|f\|_{L^\infty([0,1]^d)} + 5\sqrt{d}\,\omega_f(1).$$

Clearly, $\big\|g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1\big\|_{L^\infty(\mathbb{R}^d)} \le M$. Moreover, for any $\boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, we have

$$\begin{aligned} \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) &\in \Big[f(\boldsymbol{x}) - 5\sqrt{d}\,\omega_f(1),\ f(\boldsymbol{x}) + 5\sqrt{d}\,\omega_f(1)\Big] \\ &\subseteq \Big[-\|f\|_{L^\infty([0,1]^d)} - 5\sqrt{d}\,\omega_f(1),\ \|f\|_{L^\infty([0,1]^d)} + 5\sqrt{d}\,\omega_f(1)\Big] = [-M, M], \end{aligned}$$

implying

$$g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}).$$

We claim $g_2 \in \mathcal{NN}\{4,\ 2;\ \mathbb{R} \to \mathbb{R}\}$. To see this, we need to show how to realize g_2 by a ReLU network. Clearly, we have

$$g_2(x) + M = \min\big\{\sigma(x+M),\ 2M\big\} \quad \text{for any } x \in \mathbb{R},$$

implying

$$\begin{aligned} g_2(x) &= \min\big\{\sigma(x+M),\ 2M\big\} - M \\ &= \tfrac{1}{2}\Big(\sigma\big(\sigma(x+M)+M\big) - \sigma\big(-\sigma(x+M)-M\big) - \sigma\big(\sigma(x+M)-M\big) - \sigma\big(-\sigma(x+M)+M\big)\Big) - M, \end{aligned}$$

where the last equality comes from

$$\min\{a,b\} = \tfrac{1}{2}\big(a+b-|a-b|\big) = \tfrac{1}{2}\big(\sigma(a+b) - \sigma(-a-b) - \sigma(a-b) - \sigma(-a+b)\big) \quad \text{for any } a, b \in \mathbb{R}.$$

As shown in Figure 8, $g_2 \in \mathcal{NN}\{4,\ 2;\ \mathbb{R} \to \mathbb{R}\}$ as desired.

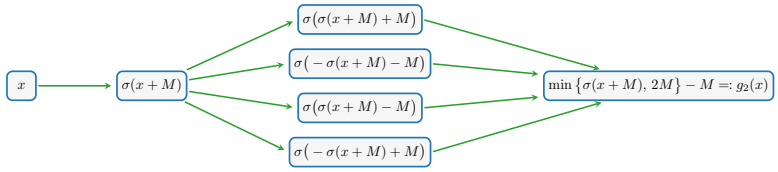


Figure 8. An illustration of the ReLU network realizing g_2.

Let $\widehat{\mathcal{L}}_3 : \mathbb{R} \to \mathbb{R}$ as the identity map. Then, by Proposition 3.3 with $N_1 = 39d + 24$, $N_2 = 4$, $L_1 = 3$, $L_2 = 2$, $d_0 = d$, $d_1 = 5d + 3$, $d_2 = d_3 = 1$ therein and setting $\widetilde{d} = 5d + 3 \ge \max\{d_1, d_2\}$, there exist

$$\begin{aligned} \boldsymbol{g} &\in \mathcal{NN}\big\{(39d+24) + 4 + 6\widetilde{d} + 2,\ \max\{3+2, 2+1\};\ \mathbb{R}^{\widetilde{d}+2} \to \mathbb{R}^{\widetilde{d}+2}\big\} \\ &= \mathcal{NN}\big\{69d + 48,\ 5;\ \mathbb{R}^{5d+5} \to \mathbb{R}^{5d+5}\big\} \end{aligned}$$

and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+5}$ and $\mathcal{L}_2 : \mathbb{R}^{5d+5} \to \mathbb{R}$ such that

$$\widehat{\mathcal{L}}_3 \circ g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r-1+1+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x})$$

for any $\boldsymbol{x} \in [-1,1]^d \supseteq [0,1]^d$. By defining $\phi \coloneqq \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1$, we have

$$\phi(\boldsymbol{x}) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) = \widehat{\mathcal{L}}_3 \circ g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x})$$

for any $\boldsymbol{x} \in [0,1]^d$. Recall that $\left\| g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1 \right\|_{L^\infty(\mathbb{R}^d)} \le M$ and

$$g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta).$$

Thus, we have

$$\left|\phi(\boldsymbol{x}) - f(\boldsymbol{x})\right| \le \|\phi\|_{L^\infty([0,1]^d)} + \|f\|_{L^\infty([0,1]^d)} \le \left\| g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1 \right\|_{L^\infty([0,1]^d)} + M \le 2M$$

for any $\boldsymbol{x} \in [0,1]^d$ and

$$\left|\phi(\boldsymbol{x}) - f(\boldsymbol{x})\right| = \left| g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) - f(\boldsymbol{x}) \right| = \left| \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) - f(\boldsymbol{x}) \right| \le 5\sqrt{d}\,\omega_f(r^{-1/d})$$

for any $\boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, where the last inequality comes from Equation (5).

Observe that the Lebesgue measure of $\Omega([0,1]^d, K, \delta)$ is bounded by $Kd\delta$. Hence, by choosing a small $\delta \in (0, \frac{1}{3K}]$ with

$$Kd\delta(2M)^p = \lfloor r^{-1/d} \rfloor d\delta(2M)^p \le \left(\omega_f(r^{-1/d})\right)^p,$$

we have

$$\begin{aligned}
\|\phi - f\|^p_{L^p([0,1]^d)} &= \int_{\Omega([0,1]^d,K,\delta)} |\phi(\boldsymbol{x}) - f(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} + \int_{[0,1]^d \backslash \Omega([0,1]^d,K,\delta)} |\phi(\boldsymbol{x}) - f(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} \\
&\le Kd\delta(2M)^p + \left(5\sqrt{d}\,\omega_f(r^{-1/d})\right)^p \\
&\le \left(\omega_f(r^{-1/d})\right)^p + \left(5\sqrt{d}\,\omega_f(r^{-1/d})\right)^p \le \left(6\sqrt{d}\,\omega_f(r^{-1/d})\right)^p.
\end{aligned}$$

Therefore, we can conclude that $\|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1 - f\|_{L^p([0,1]^d)} = \|\phi - f\|_{L^p([0,1]^d)} \le 6\sqrt{d}\,\omega_f(r^{-1/d})$. Thus, we have completed the proof of Theorem 1.1. □

A.3. Proof of Theorem 1.3 with Theorem A.1

To establish Theorem 1.3, we will rely on Theorem A.1, which permits unbounded approximation errors in the trifling region $\Omega([0,1]^d, K, \delta)$. However, when it comes to proving Theorem 1.3 using pointwise approximation, it becomes essential to control the approximation error within the trifling region. To address this, we introduce a separate theorem that specifically deals with the approximation within the trifling region.

Theorem A.2 (Lemma 3.11 of (Zhang, 2020) or Lemma 3.4 of (Lu et al., 2021))**.** *Given any $\varepsilon > 0$, $K \in \mathbb{N}^+$, and $\delta \in (0, \frac{1}{3K}]$, assume $f \in C([0,1]^d)$ and $g : \mathbb{R}^d \to \mathbb{R}$ is a general function with*

$$|g(\boldsymbol{x}) - f(\boldsymbol{x})| \le \varepsilon \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta).$$

Then

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

where $\phi \coloneqq \phi_d$ is defined by induction through $\phi_0 \coloneqq g$ and

$$\phi_{i+1}(\boldsymbol{x}) \coloneqq \mathrm{mid}\big(\phi_i(\boldsymbol{x} - \delta \boldsymbol{e}_{i+1}),\ \phi_i(\boldsymbol{x}),\ \phi_i(\boldsymbol{x} + \delta \boldsymbol{e}_{i+1})\big) \quad i = 0, 1, \cdots, d-1,$$

where $\{\boldsymbol{e}_i\}_{i=1}^d$ is the standard basis in $\mathbb{R}^d$ and $\mathrm{mid}(\cdot,\cdot,\cdot)$ is the function returning the middle value of three inputs.

Now, we are prepared to provide the detailed proof of Theorem 1.3 by assuming the validity of Theorem A.1.

Proof of Theorem 1.3. We may assume f is not a constant function since it is a trivial case. Then $\omega_f(t) > 0$ for any $t > 0$. Set $K = \lfloor r^{1/d} \rfloor$ and choose a sufficiently small $\delta \in (0, \frac{1}{3K}]$ such that

$$d \cdot \omega_f(\delta) \le \omega_f\big(r^{-1/d}\big).$$

By Theorem A.1, there exist $\boldsymbol{g}_0 \in \mathcal{NN}\big\{39d+24,\ 3;\ \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\big\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_{0,1} : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ and $\mathcal{L}_{0,2} : \mathbb{R}^{5d+3} \to \mathbb{R}$ such that

$$\big|\mathcal{L}_{0,2} \circ \boldsymbol{g}_0^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_{0,1}(\boldsymbol{x}) - f(\boldsymbol{x})\big| \le 5\sqrt{d}\,\omega_f(r^{-1/d}) \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta).$$

Define $\phi_0 \coloneqq \mathcal{L}_{0,2} \circ \boldsymbol{g}_0^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_{0,1}$. By Theorem A.2 with $g = \phi_0$ and $\varepsilon = 5\sqrt{d}\,\omega_f\big(r^{-1/d}\big) > 0$ therein, we have

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \le \varepsilon + d \cdot \omega_f(\delta) \le 6\sqrt{d}\,\omega_f\big(r^{-1/d}\big) \quad \text{for any } \boldsymbol{x} \in [0,1]^d, \tag{6}$$

where $\phi \coloneqq \phi_d$ is defined by induction through

$$\phi_{i+1}(\boldsymbol{x}) \coloneqq \text{mid}\Big(\phi_i(\boldsymbol{x} - \delta \boldsymbol{e}_{i+1}),\ \phi_i(\boldsymbol{x}),\ \phi_i(\boldsymbol{x} + \delta \boldsymbol{e}_{i+1})\Big) \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d \text{ and } i = 0, 1, \cdots, d-1.$$

Here, $\{\boldsymbol{e}_i\}_{i=1}^d$ is the standard basis in $\mathbb{R}^d$ and $\text{mid}(\cdot,\cdot,\cdot)$ is the function returning the middle value of three inputs.

It remains to show $\phi = \phi_d$ can be represented as the desired form. We claim that ϕ_i can be represented as

$$\phi_i = \mathcal{L}_{i,2} \circ \boldsymbol{g}_i^{\circ r_i} \circ \boldsymbol{\mathcal{L}}_{i,1} \quad \text{on } [-A_i, A_i]^d \quad \text{for } i = 0, 1, \cdots, d,$$

where r_i, A_i, $\boldsymbol{\mathcal{L}}_{i,1}$, $\mathcal{L}_{i,2}$, and $\boldsymbol{g}_i$ satisfy the following conditions:

- $r_i = 3r + 2i - 1$ and $A_i = d + 1 - i$;
- $\boldsymbol{\mathcal{L}}_{i,1} : \mathbb{R}^d \to \mathbb{R}^{d_i}$ and $\mathcal{L}_{i,2} : \mathbb{R}^{d_i} \to \mathbb{R}$ are two affine linear maps with $d_i = 3^i(5d+4) - 1$;
- $\boldsymbol{g}_i \in \mathcal{NN}\{N_i,\ L_i;\ \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}\}$ with $N_i = 4^{i+5}d$ and $L_i = 3 + 2i$.

We will prove this claim by induction. First, let us consider the base case $i = 0$. Clearly, $\phi_0 = \mathcal{L}_{0,2} \circ \boldsymbol{g}_0^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_{0,1} = \mathcal{L}_{0,2} \circ \boldsymbol{g}_0^{\circ r_0} \circ \boldsymbol{\mathcal{L}}_{0,1}$ on $\mathbb{R}^d \supseteq [-A_0, A_0]^d$, where $d_0 = 3^0(5d+4) - 1 = 5d+3$, $\boldsymbol{\mathcal{L}}_{0,1} : \mathbb{R}^d \to \mathbb{R}^{d_0}$ and $\mathcal{L}_{0,2} : \mathbb{R}^{d_0} \to \mathbb{R}$ are two affine linear maps and

$$\boldsymbol{g}_0 \in \mathcal{NN}\big\{39d+24,\ 3;\ \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\big\} \subseteq \mathcal{NN}\Big\{N_0 = 4^{0+5}d = 1024d,\ L_0 = 3 + 0 = 3;\ \mathbb{R}^{d_0} \to \mathbb{R}^{d_0}\Big\}.$$

Next, let us assume the claim holds for the case $i = j \in \{0, 1, \cdots, d-1\}$. We will prove the claim for the case $i = j+1$. By the induction hypothesis, ϕ_i can be represented as

$$\phi_j = \mathcal{L}_{j,2} \circ \boldsymbol{g}_j^{\circ r_j} \circ \boldsymbol{\mathcal{L}}_{j,1} \quad \text{on } [-A_j, A_j]^d \quad \text{for } j = 0, 1, \cdots, d,$$

where $\boldsymbol{\mathcal{L}}_{j,1} : \mathbb{R}^d \to \mathbb{R}^{d_j}$ and $\mathcal{L}_{j,2} : \mathbb{R}^{d_j} \to \mathbb{R}$ are two affine linear maps and $\boldsymbol{g}_j \in \mathcal{NN}\{N_j,\ L_j;\ \mathbb{R}^{d_j} \to \mathbb{R}^{d_j}\}$.

Define $\widehat{\boldsymbol{\mathcal{L}}}_{j+1,1} : \mathbb{R}^d \to \mathbb{R}^{3d_j}$ via

$$\widehat{\boldsymbol{\mathcal{L}}}_{j+1,1}(\boldsymbol{x}) \coloneqq \Big(\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x} - \delta \boldsymbol{e}_{j+1}),\ \boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}),\ \boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x} + \delta \boldsymbol{e}_{k+1})\Big) \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d,$$

$\widehat{\boldsymbol{g}}_{j+1} : \mathbb{R}^{3d_j} \to \mathbb{R}^{3d_j}$ via

$$\widehat{\boldsymbol{g}}_{j+1}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \coloneqq \Big(\boldsymbol{g}_j(\boldsymbol{u}),\ \boldsymbol{g}_j(\boldsymbol{v}),\ \boldsymbol{g}_j(\boldsymbol{w})\Big) \quad \text{for any } \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^{d_j},$$

$\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}:\mathbb{R}^{3d_j}\to\mathbb{R}^3$ via

$$\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w})\coloneqq\Big(\mathcal{L}_{j,2}(\boldsymbol{u}),\ \mathcal{L}_{j,2}(\boldsymbol{v}),\ \mathcal{L}_{j,2}(\boldsymbol{w})\Big)\quad\text{for any }\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\in\mathbb{R}^{d_j},$$

$\widehat{\boldsymbol{G}}:\mathbb{R}^3\to\mathbb{R}^3$ via

$$\widehat{\boldsymbol{G}}(y_1,y_2,y_3)\coloneqq\Big(\text{mid}(y_1,y_2,y_3),\ 0,\ 0\Big)\quad\text{for any }(y_1,y_2,y_3)\in\mathbb{R}^3,$$

and $\widehat{\mathcal{L}}_3:\mathbb{R}^3\to\mathbb{R}$ via

$$\widehat{\mathcal{L}}_3(y_1,y_2,y_3)\coloneqq y_1\quad\text{for any }(y_1,y_2,y_3)\in\mathbb{R}^3.$$

Note that $A_{j+1}=d+1-(j+1)=A_j-1\le A_j-\delta$. For any $\boldsymbol{x}\in[-A_{j+1},A_{j+1}]^d\subseteq[-A_j+\delta,A_j-\delta]^d$, we have $\boldsymbol{x}-\delta\boldsymbol{e}_{j+1},\ \boldsymbol{x},\ \boldsymbol{x}+\delta\boldsymbol{e}_{j+1}\in[-A_j,A_j]^d$, implying

$$\begin{aligned}
\phi_{j+1}(\boldsymbol{x})&=\text{mid}\Big(\phi_j(\boldsymbol{x}-\delta\boldsymbol{e}_{j+1}),\ \phi_j(\boldsymbol{x}),\ \phi_j(\boldsymbol{x}+\delta\boldsymbol{e}_{j+1})\Big)=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\Big(\phi_j(\boldsymbol{x}-\delta\boldsymbol{e}_{j+1}),\ \phi_j(\boldsymbol{x}),\ \phi_j(\boldsymbol{x}+\delta\boldsymbol{e}_{j+1})\Big)\\
&=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\Bigg(\mathcal{L}_{j,2}\circ\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}-\delta\boldsymbol{e}_{j+1}),\quad\mathcal{L}_{j,2}\circ\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}),\quad\mathcal{L}_{j,2}\circ\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}+\delta\boldsymbol{e}_{j+1})\Bigg)\\
&=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}\Bigg(\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}-\delta\boldsymbol{e}_{j+1}),\quad\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}),\quad\boldsymbol{g}_j^{\circ r_j}\circ\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}+\delta\boldsymbol{e}_{j+1})\Bigg)\\
&=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}\circ\widehat{\boldsymbol{g}}_{j+1}^{\circ r_j}\Bigg(\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}-\delta\boldsymbol{e}_{j+1}),\quad\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}),\quad\boldsymbol{\mathcal{L}}_{j,1}(\boldsymbol{x}+\delta\boldsymbol{e}_{k+1})\Bigg)\\
&=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}\circ\widehat{\boldsymbol{g}}_{j+1}^{\circ r_j}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,1}(\boldsymbol{x}).
\end{aligned}$$

Clearly, $\boldsymbol{g}_j\in\mathcal{NN}\{N_j,\ L_j;\ \mathbb{R}^{d_j}\to\mathbb{R}^{d_j}\}$ implies $\widehat{\boldsymbol{g}}_{j+1}\in\mathcal{NN}\{3N_j,\ L_j;\ \mathbb{R}^{3d_j}\to\mathbb{R}^{3d_j}\}$. By Lemma 3.1 of (Shen et al., 2021b), $\text{mid}(\cdot,\cdot,\cdot)$ can be realized by a ReLU network of width 14 and depth 2, implying $\widehat{\boldsymbol{G}}\in\mathcal{NN}\{14,\ 2;\ \mathbb{R}^3\to\mathbb{R}^3\}$.

Then, by Proposition 3.3 with $\widehat{N}_1=3N_j$, $\widehat{N}_2=14$, $\widehat{L}_1=L_j$, $\widehat{L}_2=2$, $\widehat{d}_0=d$, $\widehat{d}_1=3d_j$, $\widehat{d}_2=3$, $\widehat{d}_3=1$ therein and setting $\widehat{d}=3d_j=\max\{3d_j,3\}=\max\{\widehat{d}_1,\widehat{d}_2\}$, there exist

$$\begin{aligned}
\boldsymbol{g}_{j+1}&\in\mathcal{NN}\Big\{\widehat{N}_1+\widehat{N}_2+6\widehat{d}+2,\quad\max\{\widehat{L}_1+2,\ \widehat{L}_2+1\};\ \mathbb{R}^{\widehat{d}+2}\to\mathbb{R}^{\widehat{d}+2}\Big\}\\
&=\mathcal{NN}\Big\{3N_j+14+18d_j+2,\quad\max\{L_j+2,\ 2+1\};\ \mathbb{R}^{3d_j+2}\to\mathbb{R}^{3d_j+2}\Big\}\\
&=\mathcal{NN}\Big\{3(4^{j+5}d)+18\big(3^j(5d+4)-1\big)+16,\quad 3+2j+2;\ \mathbb{R}^{3(3^j(5d+4)-1)+2}\to\mathbb{R}^{3(3^j(5d+4)-1)+2}\Big\}\\
&\subseteq\mathcal{NN}\Big\{3(4^{j+5}d)+3^{j+5}d,\quad 3+2(j+1);\ \mathbb{R}^{3^{j+1}(5d+4)-1}\to\mathbb{R}^{3^{j+1}(5d+4)-1}\Big\}\\
&\subseteq\mathcal{NN}\Big\{4^{(j+1)+5}d,\quad 3+2(j+1);\ \mathbb{R}^{3^{j+1}(5d+4)-1}\to\mathbb{R}^{3^{j+1}(5d+4)-1}\Big\}=\mathcal{NN}\big\{N_{j+1},\ L_{j+1};\ \mathbb{R}^{d_{j+1}}\to\mathbb{R}^{d_{j+1}}\big\}
\end{aligned}$$

and two affine linear maps $\boldsymbol{\mathcal{L}}_{j+1,1}:\mathbb{R}^d\to\mathbb{R}^{d_{j+1}}$ and $\mathcal{L}_{j+1,2}:\mathbb{R}^{d_{j+1}}\to\mathbb{R}$ such that

$$\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}\circ\widehat{\boldsymbol{g}}_{j+1}^{\circ r_j}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,1}(\boldsymbol{x})=\mathcal{L}_{j+1,2}\circ\boldsymbol{g}_{j+1}^{\circ(r_j+1+1)}\circ\boldsymbol{\mathcal{L}}_{j+1,1}(\boldsymbol{x})=\mathcal{L}_{j+1,2}\circ\boldsymbol{g}_{j+1}^{\circ r_{j+1}}\circ\boldsymbol{\mathcal{L}}_{j+1,1}(\boldsymbol{x})$$

for any $\boldsymbol{x}\in[-A_{j+1},A_{j+1}]$, where the last equality comes from $r_j+1+1=(3r+2j-1)+1+1=3r+2(j+1)-1=r_{j+1}$. Therefore, for any $\boldsymbol{x}\in[-A_{j+1},A_{j+1}]$, we have

$$\phi_{j+1}(\boldsymbol{x})=\widehat{\mathcal{L}}_3\circ\widehat{\boldsymbol{G}}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,2}\circ\widehat{\boldsymbol{g}}_{j+1}^{\circ r_j}\circ\widehat{\boldsymbol{\mathcal{L}}}_{j+1,1}(\boldsymbol{x})=\mathcal{L}_{j+1,2}\circ\boldsymbol{g}_{j+1}^{\circ r_{j+1}}\circ\boldsymbol{\mathcal{L}}_{j+1,1}(\boldsymbol{x}).$$

By the principle of mathematical induction, we finish the proof of the claim.

Then, by the claim and setting $\widetilde{d}=3^d(5d+4)-1$, $\phi=\phi_d$ can be represented as

$$\phi=\phi_d=\mathcal{L}_{d,2}\circ\boldsymbol{g}_d^{\circ r_d}\circ\boldsymbol{\mathcal{L}}_{d,1}=\mathcal{L}_{d,2}\circ\boldsymbol{g}_d^{\circ(3r+2d-1)}\circ\boldsymbol{\mathcal{L}}_{d,1}\quad\text{on }[-A_d,A_d]^d=[-1,1]^d\supseteq[0,1]^d,$$

where $\boldsymbol{\mathcal{L}}_{d,1}:\mathbb{R}^d\to\mathbb{R}^{\widetilde{d}}$ and $\mathcal{L}_{i,2}:\mathbb{R}^{\widetilde{d}}\to\mathbb{R}$ are two affine linear maps and

$$\boldsymbol{g}_d\in\mathcal{NN}\{N_d,\,L_d;\;\mathbb{R}^{\widetilde{d}}\to\mathbb{R}^{\widetilde{d}}\}=\mathcal{NN}\{4^{d+5}d,\,3+2d;\;\mathbb{R}^{\widetilde{d}}\to\mathbb{R}^{\widetilde{d}}\}.$$

By defining $\boldsymbol{\mathcal{L}}_1\coloneqq\boldsymbol{\mathcal{L}}_{d,1}$, $\boldsymbol{g}\coloneqq\boldsymbol{g}_d$, and $\mathcal{L}_2\coloneqq\mathcal{L}_{d,2}$, we have $\mathcal{L}_2\circ\boldsymbol{g}^{\circ(3r+2d-1)}\circ\boldsymbol{\mathcal{L}}_1=\mathcal{L}_{d,2}\circ\boldsymbol{g}_d^{\circ(3r+2d-1)}\circ\boldsymbol{\mathcal{L}}_{d,1}=\phi$. It follows from Equation (6) that

$$\left|\mathcal{L}_2\circ\boldsymbol{g}^{\circ(3r+2d-1)}\circ\boldsymbol{\mathcal{L}}_1(\boldsymbol{x})-f(\boldsymbol{x})\right|=|\phi(\boldsymbol{x})-f(\boldsymbol{x})|\le 6\sqrt{d}\,\omega_f(r^{-1/d})\quad\text{for any }\boldsymbol{x}\in[0,1]^d.$$

Thus, we finish the proof of Theorem 1.3. □

B. Proof of Theorem A.1 with Propositions

In this section, we will provide the proof of the auxiliary theorem, Theorem A.1, by relying on Propositions 3.1, 3.2, and 3.3. The detailed proofs of these propositions can be found in Sections C, D, and E, respectively. By assuming the validity of these three propositions, we now proceed to prove Theorem A.1.

Proof of Theorem A.1. We may assume $\omega_f(t)>0$ for any $t>0$ since $\omega_f(t_0)=0$ for some $t_0>0$ implies f is a constant function, which is a trivial case. Clearly, $|f(\boldsymbol{x})-f(\boldsymbol{0})|\le\omega_f(\sqrt{d})$ for any $\boldsymbol{x}\in[0,1]^d$. By defining

$$\widetilde{f}\coloneqq f-f(\boldsymbol{0})+\omega_f(\sqrt{d}),$$

we have $\omega_{\widetilde{f}}(t)=\omega_f(t)$ for any $t\ge 0$ and $0\le\widetilde{f}(\boldsymbol{x})\le 2\omega_f(\sqrt{d})$ for any $\boldsymbol{x}\in[0,1]^d$.

Set $K=\lfloor r^{1/d}\rfloor$ and let δ be an arbitrary number in $(0,\frac{1}{3K}]$. The proof can be divided into four main steps as follows.

1. Divide $[0,1]^d$ into a set of cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and $\Omega([0,1]^d,K,\delta)$. Denote $\boldsymbol{x}_{\boldsymbol{\beta}}$ as the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm, where $\Omega([0,1]^d,K,\delta)$ is the trifling region defined in Equation (4).

2. Use Proposition 3.1 to construct a vector function $\boldsymbol{\Phi}_1=\widehat{\boldsymbol{\mathcal{L}}}_2\circ\boldsymbol{G}_1^{\circ(r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_1$ mapping $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$, i.e., $\boldsymbol{\Phi}_1(\boldsymbol{x})=\boldsymbol{\beta}$ for all $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\mathcal{L}}}_1$ and $\widehat{\boldsymbol{\mathcal{L}}}_2$ are affine linear maps and $\boldsymbol{G}_1$ is realized by a fixed-size ReLU network.

3. Construct a function $\phi_2=\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3$ mapping the index $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$, where $\widehat{\boldsymbol{\mathcal{L}}}_3$, $\widehat{\boldsymbol{\mathcal{L}}}_4$, and $\widehat{\mathcal{L}}_5$ are affine linear maps and $\boldsymbol{g}_2$ is realized by a fixed-size ReLU network. This core step can be further divided into two sub-steps:

   3.1. Design an affine linear map $\widehat{\boldsymbol{\mathcal{L}}}_3$ bijectively mapping the index set $\{0,1,\cdots,K-1\}^d$ to an auxiliary set $\mathcal{A}_1\subseteq\left\{\frac{j}{2K^d}:j=0,1,\cdots,2K^d\right\}$ defined later. See Figure 10 for an illustration.

   3.2. Apply Proposition 3.2 to design a sub-network to realize a function $\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4$ mapping $\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$. Then, $\phi_2=\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3$ maps $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$.

4. Construct the desired function ϕ via $\phi=\phi_2\circ\boldsymbol{\Phi}_1+f(\boldsymbol{0})-\omega_f(\sqrt{d})=\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3\circ\widehat{\boldsymbol{\mathcal{L}}}_2\circ\boldsymbol{G}_1^{\circ(r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_1+f(\boldsymbol{0})-\omega_f(\sqrt{d})$ and we use Proposition 3.3 to show ϕ can be represented as $\mathcal{L}_2\circ\boldsymbol{g}^{\circ(3r-1)}\circ\boldsymbol{\mathcal{L}}_1$, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are affine linear maps and $\boldsymbol{g}$ is realized by a fixed-size ReLU network. Then we have $\phi_2\circ\boldsymbol{\Phi}_1(\boldsymbol{x})=\phi_2(\boldsymbol{\beta})\approx\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\approx\widetilde{f}(\boldsymbol{x})$ for any $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$, implying $\phi(\boldsymbol{x})=\phi_2\circ\boldsymbol{\Phi}_1(\boldsymbol{x})+f(\boldsymbol{0})-\omega_f(\sqrt{d})\approx\widetilde{f}(\boldsymbol{x})+f(\boldsymbol{0})-\omega_f(\sqrt{d})=f(\boldsymbol{x})$.

The details of the above steps are presented below.

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

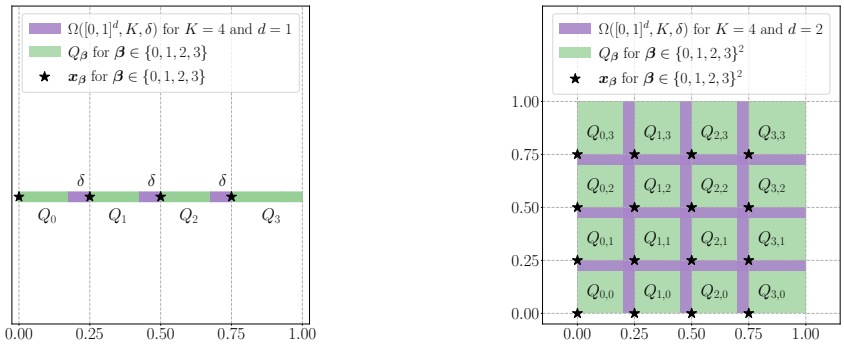
For each d-dimensional index $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$, define $\boldsymbol{x}_{\boldsymbol{\beta}} := \boldsymbol{\beta}/K$ and

$$Q_{\boldsymbol{\beta}} := \Big\{\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in [0,1]^d : x_i \in \big[\tfrac{\beta_i}{K}, \tfrac{\beta_i+1}{K} - \delta \cdot \mathbb{1}_{\{\beta_i \le K-2\}}\big], \quad i = 1, 2, \cdots, d\Big\}.$$

Clearly, $\boldsymbol{x}_{\boldsymbol{\beta}} = \boldsymbol{\beta}/K$ is the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm and

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

where $\Omega([0,1]^d, K, \delta)$ is the trifling region defined in Equation (4). See Figure 9 for illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.



(a) (b)

Figure 9. Illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. (a) $K = 4$ and $d = 1$. (b) $K = 4$ and $d = 2$.

Step 2: Construct $\boldsymbol{\Phi}_1$ mapping $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$.

By Proposition 3.1 with $m = r$ and $n = K = \lfloor r^{1/d} \rfloor \le r = m$ therein and setting $\widetilde{\delta} = K\delta$, there exist $\boldsymbol{g}_1 \in \mathcal{NN}\{9,\ 1;\ \mathbb{R}^5 \to \mathbb{R}^5\}$ and two affine linear maps $\widetilde{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R} \to \mathbb{R}^5$ and $\widetilde{\mathcal{L}}_2 : \mathbb{R}^5 \to \mathbb{R}$ such that

$$\widetilde{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(t) = k \quad \text{for any } t \in \big[k,\ k+1-\widetilde{\delta} \cdot \mathbb{1}_{\{k \le K-2\}}\big] \text{ and } k = 0, 1, \cdots, K-1.$$

Define $\boldsymbol{G}_1 : \mathbb{R}^{5d} \to \mathbb{R}^{5d}$ via

$$\boldsymbol{G}_1(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_d) = \Big(\boldsymbol{g}_1(\boldsymbol{y}_1),\ \cdots,\ \boldsymbol{g}_1(\boldsymbol{y}_d)\Big) \quad \text{for any } \boldsymbol{y}_1, \cdots, \boldsymbol{y}_d \in \mathbb{R}^5,$$

$\widehat{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d}$ via

$$\widehat{\boldsymbol{\mathcal{L}}}_1(x_1, \cdots, x_d) = \Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_1),\ \cdots,\ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_d)\Big) \quad \text{for any } (x_1, \cdots, x_d) \in \mathbb{R}^d,$$

and $\widehat{\boldsymbol{\mathcal{L}}}_2 : \mathbb{R}^{5d} \to \mathbb{R}^d$ via

$$\widehat{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_d) = \Big(\widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{y}_1),\ \cdots,\ \widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{y}_d)\Big) \quad \text{for any } \boldsymbol{y}_1, \cdots, \boldsymbol{y}_d \in \mathbb{R}^5.$$

It is easy to verify that $\boldsymbol{G}_1 \in \mathcal{NN}\{9d,\ 1;\ \mathbb{R}^{5d} \to \mathbb{R}^{5d}\}$.

For any $\boldsymbol{x} = (x_1, \cdots, x_d) \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$, we have

$$Kx_i \in \big[\beta_i,\ \beta_i + 1 - K\delta \cdot \mathbb{1}_{\{\beta_i \le K-2\}}\big] = \big[\beta_i,\ \beta_i + 1 - \widetilde{\delta} \cdot \mathbb{1}_{\{\beta_i \le K-2\}}\big]$$

for $i = 1, 2, \cdots, d$, implying

$$\widetilde{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_i) = \beta_i.$$

Therefore, for any $\boldsymbol{x} = (x_1, \cdots, x_d) \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$, we have

$$\begin{aligned}
\widehat{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{G}_1^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) &= \widehat{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{G}_1^{\circ(r-1)}\Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_1),\ \cdots,\ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_d)\Big) \\
&= \widehat{\boldsymbol{\mathcal{L}}}_2\Big(\boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_1), \quad \cdots, \quad \boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_d)\Big) \\
&= \Big(\widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_1), \quad \cdots, \quad \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(Kx_d)\Big) \\
&= (\beta_1,\ \cdots,\ \beta_d) = \boldsymbol{\beta}.
\end{aligned}$$

By defining $\boldsymbol{\Phi}_1 \coloneqq \widehat{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{G}_1^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1$, we have

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta} \quad \text{for any } \boldsymbol{x} \in Q_{\boldsymbol{\beta}} \text{ and } \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d. \tag{7}$$

Step 3: Construct ϕ_2 mapping $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$.

We will use Proposition 3.2 to construct the desired ϕ_2. To meet the requirements of applying Proposition 3.2, we first define two auxiliary sets $\mathcal{A}_1$ and $\mathcal{A}_2$ as

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

and

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

Clearly,

$$\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\} = \left\{\frac{j}{2K^d} : j = 0, 1, \cdots, 2K^d\right\} \quad \text{and} \quad \mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset.$$

See Figure 9 for an illustration of $\mathcal{A}_1$ and $\mathcal{A}_2$. Next, we further divide this step into two sub-steps.

Step 3.1: Construct $\widehat{\boldsymbol{\mathcal{L}}}_3$ bijectively mapping $\{0, 1, \cdots, K-1\}^d$ to $\mathcal{A}_1$.

Inspired by the base-K representation, we define

$$\widehat{\mathcal{L}}_3(\boldsymbol{x}) \coloneqq \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i} \quad \text{for any } \boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d. \tag{8}$$

Then $\widehat{\mathcal{L}}_3$ is a linear function bijectively mapping the index set $\{0, 1, \cdots, K-1\}^d$ to

$$\begin{aligned}
\left\{\widehat{\mathcal{L}}_3(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\right\} &= \left\{\frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i} : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\right\} \\
&= \left\{\frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \cdots, K^{d-1} - 1 \quad \text{and} \quad k = 0, 1, \cdots, K-1\right\} = \mathcal{A}_1.
\end{aligned}$$

Step 3.2: Apply Proposition 3.2 to construct a sub-network mapping $\widehat{\mathcal{L}}_3(\boldsymbol{\beta})$ approximate to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$.

Recall that

$$\left\{\widehat{\mathcal{L}}_3(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\right\} = \mathcal{A}_1$$

and

$$\left\{\frac{j}{2K^d} : j = 0, 1, \cdots, 2K^d\right\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}.$$

We will use a set of K^d+1 points

$$\Big\{\big(1,\ \widetilde{f}(\mathbf{1})\big)\Big\}\bigcup\Big\{\Big(\widehat{\mathcal{L}}_3(\boldsymbol{\beta}),\ \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\Big):\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d\Big\}\subseteq[0,1]\times\big[0,\ 2\omega_f(\sqrt{d})\big]$$

to construct a continuous piecewise linear function $h:[0,1]\to\big[0,\ 2\omega_f(\sqrt{d})\big]$, where $\mathbf{1}=(1,\cdots,1)\in\mathbb{R}^d$. Precisely, we design h by making it satisfy the following two conditions.

- First, we set $h(1)=\widetilde{f}(\mathbf{1})$ and $h\big(\widehat{\mathcal{L}}_3(\boldsymbol{\beta})\big)=\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$, where $\mathbf{1}=(1,\cdots,1)\in\mathbb{R}^d$.
- Next, we let h be linear between any two adjacent points in $\mathcal{A}_1\cup\{1\}$.

See Figure 10 for an illustration of h. Recall that $\omega_f(t)=\omega_{\widetilde{f}}(t)$ and $\omega_f(n\cdot t)\le n\cdot\omega_f(t)$ for any $n\in\mathbb{N}^+$ and $t\in[0,\infty)$. It is easy to verify that

$$\Big|h(\tfrac{j}{2K^d})-h(\tfrac{j-1}{2K^d})\Big|\le\max\Big\{\omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K}),\ \tfrac{\omega_{\widetilde{f}}(\sqrt{d})}{K}\Big\}\le\omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K})=\omega_f(\tfrac{\sqrt{d}}{K})$$

for $j=1,2,\cdots,2K^d$.

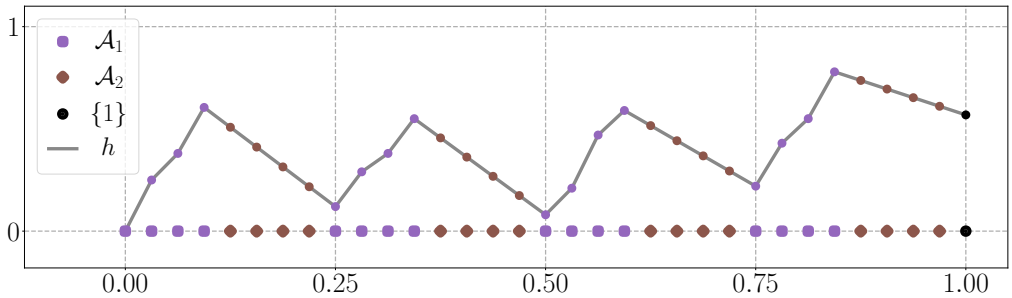


Figure 10. An illustration of $\mathcal{A}_1$, $\mathcal{A}_2$, $\{1\}$, and h for $K=4$ and $d=2$.

By Proposition 3.2 with $y_j=h(\frac{j}{2K^d})$, $\varepsilon=\omega_f(\frac{\sqrt{d}}{K})>0$, $m=2r$, and $n=2K^d=2\lfloor r^{1/d}\rfloor^d\le 2r=m$ therein, there exist $\boldsymbol{g}_2\in\mathcal{NN}\{16,\ 2;\ \mathbb{R}^6\to\mathbb{R}^6\}$ and two affine linear maps $\widetilde{\boldsymbol{\mathcal{L}}}_4:\mathbb{R}\to\mathbb{R}^6$ and $\widehat{\mathcal{L}}_5:\mathbb{R}^6\to\mathbb{R}$ such that

$$\Big|\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widetilde{\boldsymbol{\mathcal{L}}}_4(j)-h(\tfrac{j}{2K^d})\Big|\le\omega_f(\tfrac{\sqrt{d}}{K})\quad\text{for } j=0,1,\cdots,2K^d-1.$$

By defining $\widehat{\boldsymbol{\mathcal{L}}}_4(x)\coloneqq\widetilde{\boldsymbol{\mathcal{L}}}_4(2K^dx)$ for any $x\in\mathbb{R}$, we have

$$\Big|\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4(\tfrac{j}{2K^d})-h(\tfrac{j}{2K^d})\Big|=\Big|\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widetilde{\boldsymbol{\mathcal{L}}}_4(j)-h(\tfrac{j}{2K^d})\Big|\le\omega_f(\tfrac{\sqrt{d}}{K})\tag{9}$$

for $j=0,1,\cdots,2K^d-1$. Then, we can define ϕ_2 via $\phi_2\coloneqq\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3$.

By Equation (9) and $\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})\in\mathcal{A}_1\subseteq\{\frac{j}{2K^d}:j=0,1,\cdots,2K^d-1\}$ for any $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$, we have

$$\begin{aligned}\big|\phi_2(\boldsymbol{\beta})-\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\big|&=\Big|\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})-\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\Big|\\&=\Big|\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\big(\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})\big)-h\big(\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})\big)\Big|\le\omega_f(\tfrac{\sqrt{d}}{K}).\end{aligned}\tag{10}$$

Step 4: Construct the desired function ϕ and show it can be represented by the desired form.

We are ready to define the desired function ϕ via

$$\phi\coloneqq\phi_2\circ\boldsymbol{\Phi}_1+f(\mathbf{0})-\omega_f(\sqrt{d})=\widehat{\mathcal{L}}_5\circ\boldsymbol{g}_2^{\circ(2r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_4\circ\widehat{\boldsymbol{\mathcal{L}}}_3\circ\widehat{\boldsymbol{\mathcal{L}}}_2\circ\boldsymbol{G}_1^{\circ(r-1)}\circ\widehat{\boldsymbol{\mathcal{L}}}_1+f(\mathbf{0})-\omega_f(\sqrt{d}).$$

By defining $\widehat{\boldsymbol{\mathcal{L}}}_6 \coloneqq \widehat{\boldsymbol{\mathcal{L}}}_4 \circ \widehat{\boldsymbol{\mathcal{L}}}_3 \circ \widehat{\boldsymbol{\mathcal{L}}}_2$ and $\widehat{\mathcal{L}}_7 : \mathbb{R}^6 \to \mathbb{R}$ via

$$\widehat{\mathcal{L}}_7(\boldsymbol{z}) \coloneqq \widehat{\mathcal{L}}_5(\boldsymbol{z}) + f(\mathbf{0}) - \omega_f(\sqrt{d}) \quad \text{for any } \boldsymbol{z} \in \mathbb{R}^6,$$

we have $\phi = \widehat{\mathcal{L}}_7 \circ \boldsymbol{g}_2^{\circ(2r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_6 \circ \boldsymbol{G}_1^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1$.

Recall that $\boldsymbol{G}_1 \in \mathcal{NN}\{9d,\, 1;\, \mathbb{R}^{5d} \to \mathbb{R}^{5d}\}$ and $\boldsymbol{g}_2 \in \mathcal{NN}\{16,\, 2;\, \mathbb{R}^6 \to \mathbb{R}^6\}$. By Proposition 3.3 with $N_1 = 9d$, $N_2 = 16$, $L_1 = 1$, $L_2 = 2$, $d_0 = d$, $d_1 = 5d$, $d_2 = 6$ and $d_3 = 1$ therein and setting $\widetilde{d} = 5d+1 \geq \max\{5d,\, 6\}$, there exist

$$\begin{aligned}\boldsymbol{g} &\in \mathcal{NN}\{9d + 16 + 6\widetilde{d} + 2,\, \max\{1+2,\, 2+1\};\, \mathbb{R}^{\widetilde{d}+2} \to \mathbb{R}^{\widetilde{d}+2}\} \\ &= \mathcal{NN}\{39d + 24,\, 3;\, \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\}\end{aligned}$$

and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ and $\mathcal{L}_2 : \mathbb{R}^{5d+3} \to \mathbb{R}$ such that

$$\phi(\boldsymbol{x}) = \widehat{\mathcal{L}}_7 \circ \boldsymbol{g}_2^{\circ(2r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_6 \circ \boldsymbol{G}_1^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(2r-1+r-1+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x})$$

for any $\boldsymbol{x} \in [-1,1]^d \supseteq [0,1]^d$.

Next, let us estimate the approximation error. Recall that $f = \widetilde{f} + f(\mathbf{0}) - \omega_f(\sqrt{d})$ and $\phi = \phi_2 \circ \boldsymbol{\Phi}_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$. By Equations (7) and (10), for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, we have

$$\begin{aligned}\left|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) - f(\boldsymbol{x})\right| &= |\phi(\boldsymbol{x}) - f(\boldsymbol{x})| = \left|\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) - \widetilde{f}(\boldsymbol{x})\right| = \left|\phi_2(\boldsymbol{\beta}) - \widetilde{f}(\boldsymbol{x})\right| \\ &\leq \left|\phi_2(\boldsymbol{\beta}) - \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\right| + \left|\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}) - \widetilde{f}(\boldsymbol{x})\right| \\ &\leq \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_{\widetilde{f}}\big(\|\boldsymbol{x}_{\boldsymbol{\beta}} - \boldsymbol{x}\|_2\big) \leq \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K}),\end{aligned}$$

where the last inequality comes from $\|\boldsymbol{x}_{\boldsymbol{\beta}} - \boldsymbol{x}\|_2 \leq \frac{\sqrt{d}}{K}$.

Recall that $K = \lfloor r^{1/d} \rfloor \geq \frac{r^{1/d}}{2}$, $\omega_f(t) = \omega_{\widetilde{f}}(t)$, and $\omega_f(n \cdot t) \leq n \cdot \omega_f(t)$ for any $n \in \mathbb{N}^+$ and $t \in [0, \infty)$. Therefore, for any $\boldsymbol{x} \in \bigcup_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d} Q_{\boldsymbol{\beta}} = [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, we have

$$\begin{aligned}\left|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r-1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) - f(\boldsymbol{x})\right| &\leq \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K}) \leq 2\omega_f(\tfrac{\sqrt{d}}{K}) = 2\omega_f(\tfrac{\sqrt{d}}{\lfloor r^{1/d} \rfloor}) \leq 2\omega_f(2\sqrt{d}\, r^{-1/d}) \\ &\leq 2\omega_f\big(\lceil 2\sqrt{d} \rceil\, r^{-1/d}\big) \leq 2\lceil 2\sqrt{d}\, \rceil \omega_f(r^{-1/d}) \leq 5\sqrt{d}\, \omega_f(r^{-1/d}),\end{aligned}$$

where the last equality comes from the fact $2\lceil 2\sqrt{n}\, \rceil \leq 5\sqrt{n}$ for any $n \in \mathbb{N}^+$. So we finish the proof of Theorem A.1. □

C. Proof of Proposition 3.1

The main idea behind proving Proposition 3.1 lies in the composition architecture of neural networks. To streamline the proof, we begin by introducing a lemma, Lemma C.1 below, which can be seen as a weaker version of Proposition 3.1.

Lemma C.1. *Given any $\delta \in (0,1)$ and $n \in \mathbb{N}^+$ with $n \geq 2$, there exist $\boldsymbol{g} \in \mathcal{NN}\{9,\, 1;\, \mathbb{R}^5 \to \mathbb{R}^5\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R} \to \mathbb{R}^5$ and $\mathcal{L}_2 : \mathbb{R}^5 \to \mathbb{R}$ such that*

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(n-1)} \circ \boldsymbol{\mathcal{L}}_1(x) = \lfloor x \rfloor \quad \text{for any } x \in \bigcup_{\ell=0}^{n-1} \big[\ell,\, \ell + 1 - \delta\big]$$

We will prove Proposition 3.1 with Lemma C.1 in Section C.1. The proof of Lemma C.1 can be found in Section C.2.

C.1. Proof of Proposition 3.1 with Lemma C.1

Now, let us provide the detailed proof of Proposition 3.1 by assuming Lemma C.1 is true.

Proof of Proposition 3.1. We may assume $m \geq n \geq 2$ since $n = 1$ is a trivial case. Set $\widetilde{\delta} = \frac{(1-\delta)\delta}{n} \in (0,1)$. By Lemma C.1, there exist $\boldsymbol{g} \in \mathcal{NN}\{9,\, 1;\, \mathbb{R}^5 \to \mathbb{R}^5\}$ and two affine linear maps $\widetilde{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R} \to \mathbb{R}^5$ and $\mathcal{L}_2 : \mathbb{R}^5 \to \mathbb{R}$ such that

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(y) = \lfloor y \rfloor \quad \text{for any } y \in \bigcup_{k=0}^{m-1} \big[k,\, k+1-\widetilde{\delta}\big] \supseteq \bigcup_{k=0}^{n-1} \big[k,\, k+1-\widetilde{\delta}\big]. \tag{11}$$

Define $\mathcal{L}_0(x) \coloneqq \frac{n-\delta-\widetilde{\delta}}{n}x + \delta$ for any $x \in \mathbb{R}$ and $\boldsymbol{\mathcal{L}}_1 \coloneqq \widetilde{\boldsymbol{\mathcal{L}}}_1 \circ \mathcal{L}_0$. We claim

$$\mathcal{L}_0\Big(\big[k,\, k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}\big]\Big) \subseteq [k,\, k+1-\widetilde{\delta}] \quad \text{for } k=0,1,\cdots,n-1. \tag{12}$$

Then, by Equations (11) and (12), for any $x \in [k,\, k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}]$ and $k=0,1,\cdots,n-1$, we have

$$y = \mathcal{L}_0(x) \in [k,\, k+1-\widetilde{\delta}] \quad \text{for } k=0,1,\cdots,n-1,$$

from which we deduce

$$\begin{aligned}\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \boldsymbol{\mathcal{L}}_1(x) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(r-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1 \circ \mathcal{L}_0(x) &= \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1\Big(\mathcal{L}_0(x)\Big) \\ &= \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(y) = \lfloor y \rfloor = k.\end{aligned}$$

It remains to prove Equation (12). Clearly,

$$\tfrac{n-\delta-\widetilde{\delta}}{n} = \tfrac{1}{n}\big(n-\delta-\tfrac{(1-\delta)\delta}{n}\big) \ge \tfrac{1}{n}(1-\delta-(1-\delta)\delta) = \tfrac{1}{n}(1-\delta)^2 > 0,$$

implying $\mathcal{L}_0$ is increasing. To prove Equation (12), we only need to prove

$$k \le \mathcal{L}_0(k) \quad \text{for } k=0,1,\cdots,n-1 \tag{13}$$

and

$$\mathcal{L}_0\big(k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}\big) \le k+1-\widetilde{\delta} \quad \text{for } k=0,1,\cdots,n-1. \tag{14}$$

Let us first prove Equation (13). Clearly, for $k=0,1,\cdots,n-1$, we have

$$\begin{aligned}\mathcal{L}_0(k) = \tfrac{n-\delta-\widetilde{\delta}}{n}k+\delta &= k + (-\delta-\widetilde{\delta})\tfrac{k}{n} + \delta = k + \big(-\delta - \tfrac{(1-\delta)\delta}{n}\big)\tfrac{k}{n} + \delta \\ &= k + \Big(-\tfrac{k}{n} - \tfrac{(1-\delta)}{n}\tfrac{k}{n} + 1\Big)\delta = k + \tfrac{-kn-(1-\delta)k+n^2}{n^2}\delta \ge k,\end{aligned}$$

where the inequality comes from the fact $-kn-(1-\delta)k+n^2 = (n-k)n-(1-\delta)k \ge n-k \ge 0$.

Next, let us prove Equation (14). In the case of $k=n-1$, we have

$$\mathcal{L}_0\big(k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}\big) = \mathcal{L}_0(n) = \tfrac{n-\delta-\widetilde{\delta}}{n}n+\delta = n-\widetilde{\delta} = k+1-\widetilde{\delta}.$$

In the case of $k \in \{0,1,\cdots,n-2\}$, we have

$$\begin{aligned}\mathcal{L}_0\big(k+1-\delta\cdot\mathbb{1}_{\{k\le n-2\}}\big) &= \mathcal{L}_0(k+1-\delta) = \tfrac{n-\delta-\widetilde{\delta}}{n}(k+1-\delta)+\delta \\ &= \big(1-\tfrac{\delta+\widetilde{\delta}}{n}\big)(k+1-\delta)+\delta = (k+1-\delta) - \tfrac{\delta+\widetilde{\delta}}{n}(k+1-\delta)+\delta \\ &= (k+1) - \tfrac{\delta+\widetilde{\delta}}{n}(k+1-\delta) \le (k+1) - \tfrac{\delta}{n}(1-\delta) = k+1-\widetilde{\delta}.\end{aligned}$$

So we finish the proof of Proposition 3.1. □

C.2. Proof of Lemma C.1

To ensure the completeness of the proof of Proposition 3.1, we now provide the proof of Lemma C.1.

Proof of Lemma C.1. Define

$$\begin{aligned}h_k(x) &\coloneqq \sigma\big(\tfrac{k}{\delta}(x-k+\delta)\big) - \sigma\big(\tfrac{k}{\delta}(x-k)\big) + \sigma\big(\tfrac{k}{\delta}(-x+k+1)\big) - \sigma\big(\tfrac{k}{\delta}(-x+k+1-\delta)\big) - k \\ &= \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta} + k\big) - \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta}\big) + \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta}\big) - \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta} - k\big) - k\end{aligned}$$

for $k=0,1,\cdots,n-1$ and any $x \in \mathbb{R}$. See an illustration of h_k in Figure 11.

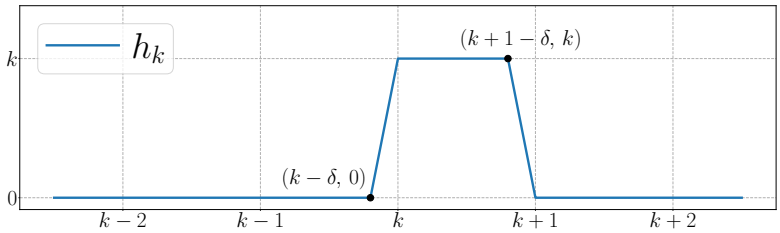


Figure 11. An illustration of h_k.

It is easy to verify that

$$h_k(x) = \begin{cases} k & \text{if } x \in [k, k+1-\delta] \\ 0 & \text{if } x \in (-\infty, k-\delta] \cup [k+1, \infty). \end{cases}$$

To see this, let us fix $k \in \{0, 1, \cdots, n-1\}$ and consider three cases below. If $x \in [k, k+1-\delta]$, we have $x - k + \delta \geq 0$, $x - k \geq 0$, $-x + k + 1 \geq 0$, and $-x + k + 1 - \delta \geq 0$, implying

$$\begin{aligned} h_k(x) &= \sigma\big(\tfrac{k}{\delta}(x-k+\delta)\big) - \sigma\big(\tfrac{k}{\delta}(x-k)\big) + \sigma\big(\tfrac{k}{\delta}(-x+k+1)\big) - \sigma\big(\tfrac{k}{\delta}(-x+k+1-\delta)\big) - k \\ &= \underbrace{\tfrac{k}{\delta}(x-k+\delta) - \tfrac{k}{\delta}(x-k)}_{=k} + \underbrace{\tfrac{k}{\delta}(-x+k+1) - \tfrac{k}{\delta}(-x+k+1-\delta)}_{=k} - k = k. \end{aligned}$$

If $x \in (-\infty, k-\delta]$, we have $x - k + \delta \leq 0$, $x - k \leq 0$, $-x + k + 1 \geq 0$, and $-x + k + 1 - \delta \geq 0$, implying

$$\begin{aligned} h_k(x) &= \sigma\big(\tfrac{k}{\delta}(x-k+\delta)\big) - \sigma\big(\tfrac{k}{\delta}(x-k)\big) + \sigma\big(\tfrac{k}{\delta}(-x+k+1)\big) - \sigma\big(\tfrac{k}{\delta}(-x+k+1-\delta)\big) - k \\ &= 0 - 0 + \underbrace{\tfrac{k}{\delta}(-x+k+1) - \tfrac{k}{\delta}(-x+k+1-\delta)}_{=k} - k = 0. \end{aligned}$$

If $x \in [k+1, \infty)$, we have $x - k + \delta \geq 0$, $x - k \geq 0$, $-x + k + 1 \leq 0$, and $-x + k + 1 - \delta \leq 0$, implying

$$\begin{aligned} h_k(x) &= \sigma\big(\tfrac{k}{\delta}(x-k+\delta)\big) - \sigma\big(\tfrac{k}{\delta}(x-k)\big) + \sigma\big(\tfrac{k}{\delta}(-x+k+1)\big) - \sigma\big(\tfrac{k}{\delta}(-x+k+1-\delta)\big) - k \\ &= \underbrace{\tfrac{k}{\delta}(x-k+\delta) - \tfrac{k}{\delta}(x-k)}_{=k} + 0 - 0 - k = 0. \end{aligned}$$

Obviously, for any $x \in [k, k+1-\delta]$ and $k = 0, 1, \cdots, n-1$, we have

$$\sum_{i=0}^{n-1} h_i(x) = h_k(x) = k = \lfloor x \rfloor.$$

It remains to construct $\boldsymbol{g}$, $\boldsymbol{\mathcal{L}}_1$, and $\mathcal{L}_2$ such that

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ n} \circ \boldsymbol{\mathcal{L}}_1(x) = \sum_{i=0}^{n-1} h_i(x) \quad \text{for any } x \in \bigcup_{i=0}^{n-1} [i,\ i+1-\delta].$$

By defining $h : \mathbb{R}^3 \to \mathbb{R}$ via

$$h(x_1, x_2, x_3) \coloneqq \sigma\big(\tfrac{x_1}{\delta} - \tfrac{x_2}{\delta} + x_3\big) - \sigma\big(\tfrac{x_1}{\delta} - \tfrac{x_2}{\delta}\big) + \sigma\big(-\tfrac{x_1}{\delta} + \tfrac{x_2}{\delta} + \tfrac{x_3}{\delta}\big) - \sigma\big(-\tfrac{x_1}{\delta} + \tfrac{x_2}{\delta} + \tfrac{x_3}{\delta} - x_3\big) - \sigma(x_3),$$

we have

$$h\big(kx, k^2, k\big) = \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta} + k\big) - \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta}\big) + \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta}\big) - \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta} - k\big) - k = h_k(x) \quad (15)$$

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

Now we are ready to construct $\boldsymbol{g}:\mathbb{R}^5\to\mathbb{R}^5$. Define

$$\boldsymbol{g}(x_1,x_2,x_3,x_4,x_5) \coloneqq \left(\sigma(x_1+x_4),\quad \sigma(x_2+2x_3+1),\quad \sigma(x_3)+1,\quad \sigma(x_4),\quad \sigma(x_5)+h(x_1,x_2,x_3)\right),$$

for any $(y_1,y_2,y_3,y_4,y_5)\in\mathbb{R}^5$, where

$$h(x_1,x_2,x_3)=\sigma\big(\tfrac{x_1}{\delta}-\tfrac{x_2}{\delta}+x_3\big)-\sigma\big(\tfrac{x_1}{\delta}-\tfrac{x_2}{\delta}\big)+\sigma\big(-\tfrac{x_1}{\delta}+\tfrac{x_2}{\delta}+\tfrac{x_3}{\delta}\big)-\sigma\big(-\tfrac{x_1}{\delta}+\tfrac{x_2}{\delta}+\tfrac{x_3}{\delta}-x_3\big)-\sigma(x_3).$$

See an illustration of the ReLU network realizing $\boldsymbol{g}:\mathbb{R}^5\to\mathbb{R}^5$ in Figure 12. Clearly, $\boldsymbol{g}\in\mathcal{NN}\{9,\ 1;\ \mathbb{R}^5\to\mathbb{R}^5\}$.

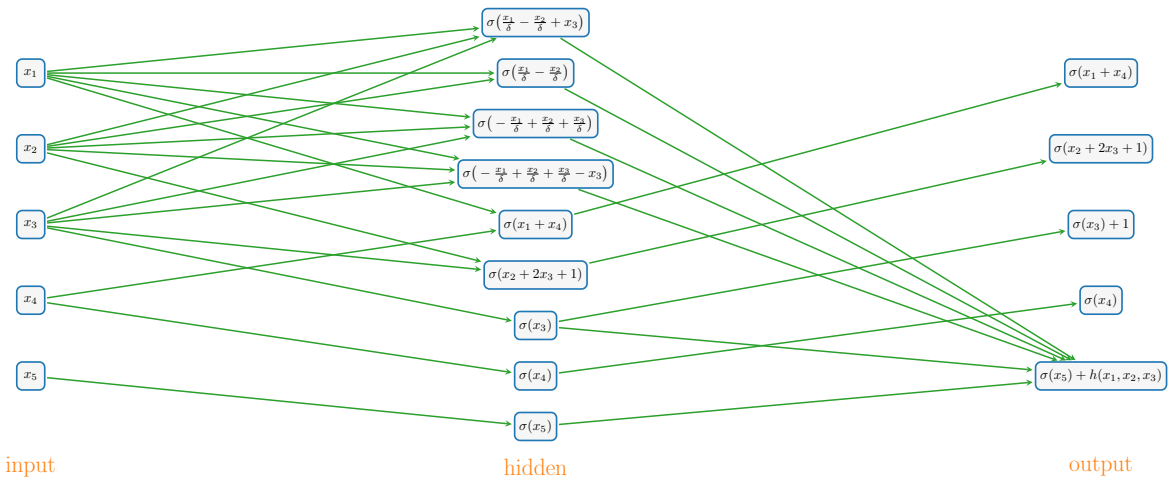


Figure 12. An illustration of $\boldsymbol{g}:\mathbb{R}^5\to\mathbb{R}^5$.

Fix $x\in\cup_{i=0}^{n-1}[i,\ i+1-\delta]$ and set

$$\boldsymbol{\xi}_k=\boldsymbol{\xi}_k(x)=\left(kx,\ k^2,\ k,\ x,\ \sum_{i=0}^{k-1}h_i(x)\right)\in[0,\infty)^5\quad\text{for } k=1,2,\cdots,n.$$

For $k=1,2,\cdots,n-1$, we have

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{\xi}_k)&=\boldsymbol{g}\left(kx,\ k^2,\ k,\ x,\ \sum_{i=0}^{k-1}h_i(x)\right)\\&=\left(\sigma(kx+x),\quad \sigma(k^2+2k+1),\quad \sigma(k)+1,\quad \sigma(x),\quad \sigma\left(\sum_{i=0}^{k-1}h_i(x)\right)+\underbrace{h(kx,k^2,k)}_{=h_k(x)\text{ by (15)}}\right)\\&=\left((k+1)x,\quad (k+1)^2,\quad (k+1),\quad x,\quad \sum_{i=0}^{(k+1)-1}h_i(x)\right)=\boldsymbol{\xi}_{k+1},\end{aligned}$$

implying $\boldsymbol{\xi}_n=\boldsymbol{g}(\boldsymbol{\xi}_{n-1})=\cdots=\boldsymbol{g}^{\circ(n-1)}(\boldsymbol{\xi}_1)$.

Define $\boldsymbol{\mathcal{L}}_1:\mathbb{R}\to\mathbb{R}^5$ via $\boldsymbol{\mathcal{L}}_1(x)\coloneqq(x,1,1,x,0)$ and $\mathcal{L}_2:\mathbb{R}^5\to\mathbb{R}$ via $\mathcal{L}_2(x_1,x_2,x_3,x_4,x_5)\coloneqq x_5$. Then, we have

$$\boldsymbol{\xi}_1=\big(x,1,1,x,h_0(x)\big)=(x,1,1,x,0)=\boldsymbol{\mathcal{L}}_1(x),$$

from which we deduce

$$\begin{aligned}\mathcal{L}_2\circ\boldsymbol{g}^{\circ(n-1)}\circ\boldsymbol{\mathcal{L}}_1(x)&=\mathcal{L}_2\circ\boldsymbol{g}^{\circ(n-1)}(\boldsymbol{\xi}_1)\\&=\mathcal{L}_2(\boldsymbol{\xi}_n)=\big[\boldsymbol{\xi}_n\big]_{[5]}=\sum_{i=0}^{n-1}h_i(x).\end{aligned}$$

So we finish the proof of Lemma C.1. □

D. Proof of Proposition 3.2

The bit extraction technique proposed in (Bartlett et al., 1998) plays a crucial role in proving Proposition 3.2. Before we delve into the proof of Proposition 3.2, we first establish Lemma D.1, which serves as a key intermediate step in the proof of Proposition 3.2.

Lemma D.1. *Given any $r \in \mathbb{N}^+$, there exist $\boldsymbol{g} \in \mathcal{NN}\{8,\ 2;\ \mathbb{R}^3 \to \mathbb{R}^3\}$ and two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^2 \to \mathbb{R}^5$ and $\mathcal{L}_2 : \mathbb{R}^5 \to \mathbb{R}$ such that: For any $\theta_1, \theta_2, \cdots, \theta_r \in \{0, 1\}$, it holds that*

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1\big(k,\ \text{bin}\,0.\theta_1\theta_2\cdots\theta_r\big) = \sum_{\ell=1}^{k} \theta_\ell \quad \text{for } k = 0, 1, \cdots, r. \tag{16}$$

We will prove Proposition 3.2 by assuming the validity of Lemma D.1, which will be proved later in Section D.2.

D.1. Proof of Proposition 3.2 with Lemma D.1

Now we are ready to give the proof of Proposition 3.2 by assuming Lemma D.1 is true.

Proof of Proposition 3.2. We may assume $n = m$ since we can set $y_{n-1} = y_n = \cdots = y_{m-1}$ if $n < m$. Set

$$a_i = \left\lfloor \tfrac{y_i}{\varepsilon} \right\rfloor \quad \text{for } i = 0, 1, \cdots, n-1$$

and

$$b_i = a_i - a_{i-1} \quad \text{for } i = 1, 2, \cdots, n-1.$$

Since $|y_i - y_{i-1}| \le \varepsilon$ for $i = 1, 2, \cdots, n-1$, we have $y_i \in [y_{i-1} - \varepsilon,\ y_{i-1} + \varepsilon]$. Thus, for $i = 1, 2, \cdots, n-1$, we have

$$-1 = \left\lfloor \tfrac{y_{i-1}-\varepsilon}{\varepsilon} \right\rfloor - \left\lfloor \tfrac{y_{i-1}}{\varepsilon} \right\rfloor \le \left\lfloor \tfrac{y_i}{\varepsilon} \right\rfloor - \left\lfloor \tfrac{y_{i-1}}{\varepsilon} \right\rfloor \le \left\lfloor \tfrac{y_{i-1}+\varepsilon}{\varepsilon} \right\rfloor - \left\lfloor \tfrac{y_{i-1}}{\varepsilon} \right\rfloor = 1,$$

implying

$$b_i = a_i - a_{i-1} = \left\lfloor \tfrac{y_i}{\varepsilon} \right\rfloor - \left\lfloor \tfrac{y_{i-1}}{\varepsilon} \right\rfloor \in [-1, 1].$$

It follows from $b_i = a_i - a_{i-1} \in \mathbb{Z}$ that $b_i \in \{-1, 0, 1\}$ for $i = 1, 2, \cdots, n-1$. Hence, there exist $c_i \in \{0, 1\}$ and $d_i \in \{0, 1\}$ such that

$$b_i = c_i - d_i \quad \text{for } i = 1, 2, \cdots, n-1.$$

Then, for any $k \in \{1, 2, \cdots, n-1\}$, we have

$$\begin{aligned} a_k = a_0 + \sum_{i=1}^{k}(a_i - a_{i-1}) &= a_0 + \sum_{i=1}^{k}(a_i - a_{i-1}) \\ &= a_0 + \sum_{i=1}^{k} b_i = a_0 + \sum_{i=1}^{k} c_i - \sum_{i=1}^{k} d_i. \end{aligned}$$

Clearly, $a_0 = a_0 + 0 - 0 = a_0 + \sum_{i=1}^{0} c_i - \sum_{i=1}^{0} d_i$. Thus, we have

$$a_k = a_0 + \sum_{i=1}^{k} c_i - \sum_{i=1}^{k} d_i \quad \text{for } k = 0, 1, \cdots, n-1.$$

By Lemma D.1 with $r = n-1$ therein, there exist $\widetilde{\boldsymbol{g}}, \widehat{\boldsymbol{g}} \in \mathcal{NN}\{8,\ 2;\ \mathbb{R}^3 \to \mathbb{R}^3\}$ and four affine linear maps $\widetilde{\boldsymbol{\mathcal{L}}}_1, \widehat{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^2 \to \mathbb{R}^3$ and $\widetilde{\mathcal{L}}_2, \widehat{\mathcal{L}}_2 : \mathbb{R}^3 \to \mathbb{R}$ such that

$$\widetilde{\mathcal{L}}_2 \circ \widetilde{\boldsymbol{g}}^{\circ(n-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1\big(k,\ \text{bin}\,0.c_1\cdots c_{n-1}\big) = \sum_{i=1}^{k} c_i \quad \text{and} \quad \widehat{\mathcal{L}}_2 \circ \widehat{\boldsymbol{g}}^{\circ(n-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1\big(k,\ \text{bin}\,0.d_1\cdots d_{n-1}\big) = \sum_{i=1}^{k} d_i$$

for $k = 0, 1, \cdots, n-1$, implying

$$\begin{aligned} a_k &= a_0 + \sum_{i=1}^{k} c_i - \sum_{i=1}^{k} d_i \\ &= a_0 + \widetilde{\mathcal{L}}_2 \circ \widetilde{\boldsymbol{g}}^{\circ(n-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1\big(k,\ \text{bin}\,0.c_1 \cdots c_{n-1}\big) - \widehat{\mathcal{L}}_2 \circ \widehat{\boldsymbol{g}}^{\circ(n-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1\big(k,\ \text{bin}\,0.d_1 \cdots d_{n-1}\big). \end{aligned} \tag{17}$$

Define $\boldsymbol{g} : \mathbb{R}^6 \to \mathbb{R}^6$ via

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \Big(\widetilde{\boldsymbol{g}}(\boldsymbol{x}),\ \widehat{\boldsymbol{g}}(\boldsymbol{y})\Big) \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3,$$

$\boldsymbol{\mathcal{L}}_1 : \mathbb{R} \to \mathbb{R}^6$ via

$$\boldsymbol{\mathcal{L}}_1(x) \coloneqq \Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(x,\ \text{bin}\,0.c_1 \cdots c_{n-1}), \quad \widehat{\boldsymbol{\mathcal{L}}}_1(x,\ \text{bin}\,0.d_1 \cdots d_{n-1})\Big) \quad \text{for any } x \in \mathbb{R},$$

and $\mathcal{L}_2 : \mathbb{R}^6 \to \mathbb{R}$ via

$$\mathcal{L}_2(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \varepsilon\Big(a_0 + \widetilde{\mathcal{L}}_2(\boldsymbol{x}) - \widehat{\mathcal{L}}_2(\boldsymbol{y})\Big) \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3.$$

It is easy to verify that $\boldsymbol{g} \in \mathcal{NN}\{16,\ 2;\ \mathbb{R}^6 \to \mathbb{R}^6\}$. Moreover, we have

$$\boldsymbol{g}^{\circ(n-1)}(\boldsymbol{x}, \boldsymbol{y}) = \Big(\widetilde{\boldsymbol{g}}^{\circ(n-1)}(\boldsymbol{x}),\ \widehat{\boldsymbol{g}}^{\circ(n-1)}(\boldsymbol{y})\Big) \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3.$$

Therefore, for $k = 0, 1, \cdots, n-1$, we have

$$\begin{aligned} \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(n-1)} \circ \boldsymbol{\mathcal{L}}_1(k) &= \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(n-1)}\Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.c_1 \cdots c_{n-1}), \quad \widehat{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.d_1 \cdots d_{n-1})\Big) \\ &= \mathcal{L}_2\Big(\widetilde{\boldsymbol{g}}^{\circ(n-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.c_1 \cdots c_{n-1}), \quad \widehat{\boldsymbol{g}}^{\circ(n-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.d_1 \cdots d_{n-1})\Big) \\ &= \varepsilon\Big(a_0 + \widetilde{\mathcal{L}}_2 \circ \widetilde{\boldsymbol{g}}^{\circ(n-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.c_1 \cdots c_{n-1}) - \widehat{\mathcal{L}}_2 \circ \widehat{\boldsymbol{g}}^{\circ(n-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(k,\ \text{bin}\,0.d_1 \cdots d_{n-1})\Big) \\ &= \varepsilon a_k, \end{aligned}$$

where the last equality comes from Equation (17). It follows that, for $k = 0, 1, \cdots, n-1$,

$$\Big|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(n-1)} \circ \boldsymbol{\mathcal{L}}_1(k) - y_k\Big| = \big|\varepsilon a_k - y_k\big| = \Big|\varepsilon \lfloor \tfrac{y_k}{\varepsilon} \rfloor - \varepsilon \tfrac{y_k}{\varepsilon}\Big| = \varepsilon \Big|\lfloor \tfrac{y_k}{\varepsilon} \rfloor - \tfrac{y_k}{\varepsilon}\Big| \le \varepsilon.$$

So we finish the proof of Proposition 3.2. □

D.2. Proof of Lemma D.1

To make the proof of Proposition 3.2 complete, we now provide the proof of Lemma D.1.

Proof of Lemma D.1. Set $\delta = 2^{-r}$ and define

$$\mathcal{T}(x) = \sigma\big(\tfrac{x}{\delta} + 1\big) - \sigma\big(\tfrac{x}{\delta}\big) \quad \text{for any } x \in \mathbb{R}.$$

See an illustration of $\mathcal{T}$ in Figure 13.

For any $\theta_1, \theta_2, \cdots, \theta_r \in \{0, 1\}$, set

$$\beta_i = \text{bin}\,0.\theta_i \cdots \theta_r \quad \text{for } i = 1, 2, \cdots, r.$$

It is easy to verify that

$$\theta_i = \mathcal{T}\big(\text{bin}\,0.\theta_i \cdots \theta_r - \tfrac{1}{2}\big) = \mathcal{T}\big(\beta_i - \tfrac{1}{2}\big) \quad \text{for } i = 1, 2, \cdots, r,$$

implying

$$\beta_{i+1} = 2\beta_i - \theta_i = 2\beta_i - \mathcal{T}\big(\beta_i - \tfrac{1}{2}\big) \quad \text{for } i = 1, 2, \cdots, r-1.$$

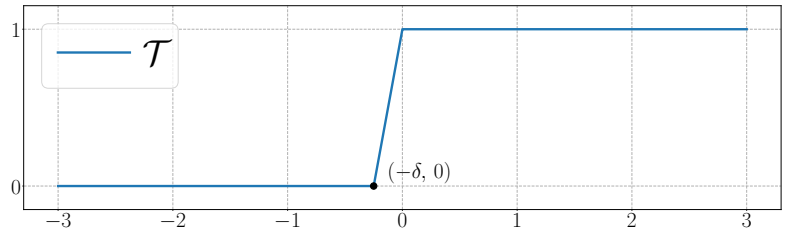


Figure 13. An illustration of $\mathcal{T}$.

By setting $\beta_{r+1} = 2\beta_r - \mathcal{T}(\beta_r - \frac{1}{2}) = 0$, we have

$$\beta_{i+1} = 2\beta_i - \mathcal{T}\big(\beta_i - \tfrac{1}{2}\big) \quad \text{for } i = 1, 2, \cdots, r.$$

Fix $k \in \{0, 1, \cdots, r\}$. The fact that $xy = \max\{0,\, x+y-1\} = \sigma(x+y-1)$ for any $x, y \in \{0, 1\}$ implies

$$\begin{aligned}\sum_{i=1}^{k} \theta_i = \sum_{i=1}^{k} \theta_i + \sum_{i=k+1}^{r} 0 = \sum_{i=1}^{r} \theta_i \cdot \mathcal{T}(k-i) &= \sum_{i=1}^{r} \sigma\Big(\theta_i + \mathcal{T}(k-i) - 1\Big) \\ &= \sum_{i=1}^{r} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big).\end{aligned} \tag{18}$$

Define $\boldsymbol{g} : \mathbb{R} \times [0, \infty)^2 \to \mathbb{R}^3$ via

$$\boldsymbol{g}(x_1, x_2, x_3) \coloneqq \left(x_1 - 1, \quad 2x_2 - \mathcal{T}(x_2 - \tfrac{1}{2}), \quad \sigma\Big(\mathcal{T}(x_2 - \tfrac{1}{2}) + \mathcal{T}(x_1) - 1\Big) + x_3\right)$$

for any $(x_1, x_2, x_3) \in \mathbb{R} \times [0, \infty)^2$.

For $\ell = 1, 2, \cdots, r+1$, we set

$$\boldsymbol{\xi}_\ell = \left(k - \ell, \quad \beta_\ell, \quad \sum_{i=1}^{\ell-1} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big)\right) \in \mathbb{R} \times [0, \infty)^2.$$

Then, for $\ell = 1, 2, \cdots, r$, we have

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{\xi}_\ell) &= \boldsymbol{g}\left(k - \ell, \quad \beta_\ell, \quad \sum_{i=1}^{\ell-1} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big)\right) \\ &= \left((k-\ell) - 1, \quad 2\beta_\ell - \mathcal{T}(\beta_\ell - \tfrac{1}{2}), \quad \sigma\Big(\mathcal{T}(\beta_\ell - \tfrac{1}{2}) + \mathcal{T}(k-\ell) - 1\Big) + \sum_{i=1}^{\ell-1} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big)\right) \\ &= \left(k - (\ell+1), \quad \beta_{\ell+1}, \quad \sum_{i=1}^{(\ell+1)-1} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big)\right) = \boldsymbol{\xi}_{\ell+1},\end{aligned}$$

implying $\boldsymbol{\xi}_{r+1} = \boldsymbol{g}(\boldsymbol{\xi}_r) = \cdots = \boldsymbol{g}^{\circ r}(\boldsymbol{\xi}_1)$.

Define $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^2 \to \mathbb{R}^3$ via $\boldsymbol{\mathcal{L}}_1(x_1, x_2) \coloneqq (x_1 - 1, x_2, 0)$ and $\mathcal{L}_2 : \mathbb{R}^3 \to \mathbb{R}$ via $\mathcal{L}_2(x_1, x_2, x_3) \coloneqq x_3$ for any $x_1, x_2, x_3 \in \mathbb{R}$. Then, we have

$$\boldsymbol{\xi}_1 = \left(k - 1,\ \beta_1,\ \sum_{i=1}^{0} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big)\right) = \Big(k - 1,\ \text{bin}\, 0.\theta_1 \cdots \theta_r,\ 0\Big) = \boldsymbol{\mathcal{L}}_1\big(k,\ \text{bin}\, 0.\theta_1 \cdots \theta_r\big),$$

from which we deduce

$$\begin{aligned}\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1(k,\, \mathrm{bin}\, 0.\theta_1 \cdots \theta_r) &= \mathcal{L}_2 \circ \boldsymbol{g}^{\circ r}(\boldsymbol{\xi}_1) = \mathcal{L}_2(\boldsymbol{\xi}_{r+1}) = \left[\boldsymbol{\xi}_{r+1}\right]_{[3]} \\ &= \sum_{i=1}^{(r+1)-1} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big) = \sum_{i=1}^{k} \theta_i,\end{aligned}$$

where the last equality comes from Equation (18). Furthermore, $\boldsymbol{g}$, $\boldsymbol{\mathcal{L}}_1$, and $\mathcal{L}_2$ are independent of $\theta_1, \cdots, \theta_r$.

It remains to show $\boldsymbol{g}$ can be realized by a ReLU network with the desired size.

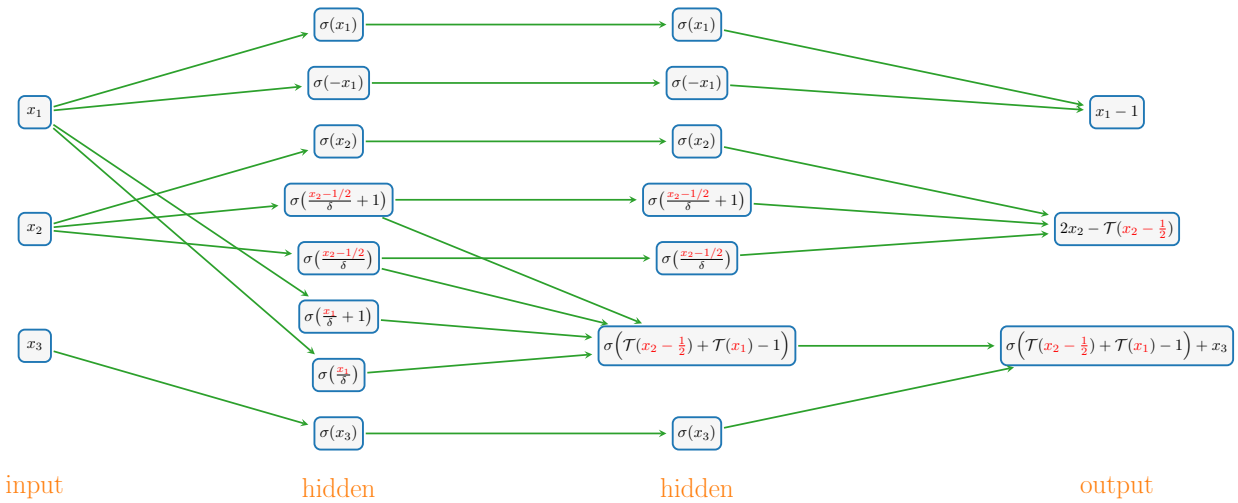


Figure 14. An illustration of the ReLU network realizing $\boldsymbol{g}(x_1, x_2, x_3)$ for $(x_1, x_2, x_3) \in \mathbb{R}\times[0,\infty)^2$ based on $\mathcal{T}(t) = \sigma\big(\frac{t}{\delta}+1\big) - \sigma\big(\frac{t}{\delta}\big)$ and $t = \sigma(t) - \sigma(-t)$ for any $t \in \mathbb{R}$.

As shown in Figure 14, $\boldsymbol{g}(x_1, x_2, x_3)$ can be realized by a ReLU network of width 8 and depth 2 for $(x_1, x_2, x_3) \in \mathbb{R} \times [0,\infty)^2$. That means, $\boldsymbol{g} \in \mathcal{NN}\{8,\, 2;\ \mathbb{R}^3 \to \mathbb{R}^3\}$. So we finish the proof of Lemma D.1. □

E. Proof of Proposition 3.3

The objective of this section is to prove Proposition 3.3. To facilitate the proof, we introduce the following lemma.

Lemma E.1. *For any $A > 0$, $\boldsymbol{g}_i \in \mathcal{NN}\big\{N_i,\, L_i;\ \mathbb{R}^d \to \mathbb{R}^d\big\}$ and $r_i \in \mathbb{N}^+$ for $i = 1, 2$, there exists*

$$\boldsymbol{\Phi} \in \mathcal{NN}\Big\{N_1 + N_2 + 2d,\, \max\{L_1, L_2\} + 1;\ \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}\Big\}$$

such that

$$\boldsymbol{\Phi}^{\circ(r_1+r_2)}(\boldsymbol{x},\, 2r_1+1) = \Big(\boldsymbol{g}_2^{\circ r_2} \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x}),\ -2r_2+1\Big)$$

for any $\boldsymbol{x} \in [-A, A]^d$.

We will prove Proposition 3.3 with Lemma E.1, which will be proved later in Section E.2.

E.1. Proof of Proposition 3.3 with Lemma E.1

Now, let us present the proof of Proposition 3.3 by assuming Lemmas E.1 is true.

Proof of Proposition 3.3. Set

$$\widetilde{A} = 100(r_1 + r_2 + 1) + \sup_{\boldsymbol{x}\in[-A,A]^{d_0}} \big\|\widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x})\big\|_{\ell^\infty}.$$

Since $d \geq \max\{d_1, d_2\}$, we can define $\widehat{\boldsymbol{g}}_1 : \mathbb{R}^d \to \mathbb{R}^d$ via

$$\widehat{\boldsymbol{g}}_1(\boldsymbol{u}, \boldsymbol{v}) \coloneqq \Big(\boldsymbol{g}_1(\boldsymbol{u}),\ \boldsymbol{0}\Big) \in \mathbb{R}^d \quad \text{for any } \boldsymbol{u} \in \mathbb{R}^{d_1} \text{ and } \boldsymbol{v} \in \mathbb{R}^{d-d_1}$$

and $\widehat{\boldsymbol{g}} : \mathbb{R}^d \to \mathbb{R}^d$ via

$$\widehat{\boldsymbol{g}}(\boldsymbol{u}, \boldsymbol{v}) \coloneqq \Big(\widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{u}),\ \boldsymbol{0}\Big) \in \mathbb{R}^d \quad \text{for any } \boldsymbol{u} \in \mathbb{R}^{d_1} \text{ and } \boldsymbol{v} \in \mathbb{R}^{d-d_1}.$$

Clearly, $\boldsymbol{g}_1 \in \mathcal{NN}\{N_1,\ L_1;\ \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}\}$ implies $\widehat{\boldsymbol{g}}_1 \in \mathcal{NN}\{N_1,\ L_1;\ \mathbb{R}^{d} \to \mathbb{R}^{d}\}$. Note that $\widetilde{\boldsymbol{\mathcal{L}}}_2 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ can be represented as

$$\widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{u}) = \sigma\Big(\widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{u})\Big) - \sigma\Big(-\widetilde{\boldsymbol{\mathcal{L}}}_2(\boldsymbol{u})\Big) \quad \text{for any } \boldsymbol{u} \in \mathbb{R}^{d_1},$$

which means $\widetilde{\boldsymbol{\mathcal{L}}}_2$ can be realized by a one-hidden-layer ReLU network of width $2d_2$. Thus, $\widehat{\boldsymbol{g}} \in \mathcal{NN}\{2d_2,\ 1;\ \mathbb{R}^d \to \mathbb{R}^d\}$.

By Lemma E.1, there exists $\boldsymbol{G}_1 \in \mathcal{NN}\{N_1 + 2d_2 + 2d,\ L_1 + 1;\ \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}\}$ such that

$$\boldsymbol{G}_1^{\circ(r_1+1)}(\boldsymbol{w},\ 2r_1 + 1) = \Big(\widehat{\boldsymbol{g}} \circ \widehat{\boldsymbol{g}}_1^{\circ r_1}(\boldsymbol{w}),\ -1\Big) \quad \text{for any } \boldsymbol{w} \in [-\widetilde{A}, \widetilde{A}]^d.$$

Define $\widehat{\boldsymbol{g}}_2 : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ via

$$\widehat{\boldsymbol{g}}_2(\boldsymbol{u}, \boldsymbol{v}) \coloneqq \Big(\boldsymbol{g}_2(\boldsymbol{u}),\ \boldsymbol{0}\Big) \in \mathbb{R}^{d+1} \quad \text{for any } \boldsymbol{u} \in \mathbb{R}^{d_2} \text{ and } \boldsymbol{v} \in \mathbb{R}^{d+1-d_2}.$$

Clearly, $\boldsymbol{g}_2 \in \mathcal{NN}\{N_2,\ L_2;\ \mathbb{R}^{d_2} \to \mathbb{R}^{d_2}\}$ implies $\widehat{\boldsymbol{g}}_2 \in \mathcal{NN}\{N_2,\ L_2;\ \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}\}$.

By Lemma E.1, there exists

$$\begin{aligned}
\boldsymbol{g} &\in \mathcal{NN}\Big\{(N_1 + 2d_2 + 2d) + N_2 + 2(d+1),\ \max\{L_1 + 1, L_2\} + 1;\ \mathbb{R}^{d+2} \to \mathbb{R}^{d+2}\Big\} \\
&\subseteq \mathcal{NN}\Big\{N_1 + N_2 + 6d + 2,\ \max\{L_1 + 2, L_2 + 1\};\ \mathbb{R}^{d+2} \to \mathbb{R}^{d+2}\Big\}
\end{aligned}$$

such that

$$\boldsymbol{g}^{\circ(r_1+1+r_2)}\Big(\boldsymbol{z},\ 2(r_1+1)+1\Big) = \Big(\widehat{\boldsymbol{g}}_2^{\circ r_2} \circ \boldsymbol{G}_1^{\circ(r_1+1)}(\boldsymbol{z}),\ -2r_2 + 1\Big) \quad \text{for any } \boldsymbol{z} \in [-\widetilde{A}, \widetilde{A}]^{d+1},$$

implying

$$\Big[\boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\boldsymbol{z},\ 2r_1 + 3\Big)\Big]_{[1:d+1]} = \widehat{\boldsymbol{g}}_2^{\circ r_2} \circ \boldsymbol{G}_1^{\circ(r_1+1)}(\boldsymbol{z}) \quad \text{for any } \boldsymbol{z} \in [-\widetilde{A}, \widetilde{A}]^{d+1}.$$

Therefore, for any $\boldsymbol{y} \in [-\widetilde{A}, \widetilde{A}]^{d_1}$, we have

$$\begin{aligned}
\Big[\boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\underbrace{\boldsymbol{y},\ \boldsymbol{0},\ 2r_1+1}_{\in[-\widetilde{A},\widetilde{A}]^{d+1}},\ 2r_1+3\Big)\Big]_{[1:d+1]} &= \widehat{\boldsymbol{g}}_2^{\circ r_2} \circ \boldsymbol{G}_1^{\circ(r_1+1)}\big(\boldsymbol{y},\ \boldsymbol{0},\ 2r_1+1\big) \\
&= \widehat{\boldsymbol{g}}_2^{\circ r_2}\Big(\widehat{\boldsymbol{g}} \circ \widehat{\boldsymbol{g}}_1^{\circ r_1}(\boldsymbol{y},\ \boldsymbol{0}),\ -1\Big) = \widehat{\boldsymbol{g}}_2^{\circ r_2}\Big(\widehat{\boldsymbol{g}}\Big(\boldsymbol{g}_1^{\circ r_1}(\boldsymbol{y}),\ \boldsymbol{0}\Big),\ -1\Big) \\
&= \widehat{\boldsymbol{g}}_2^{\circ r_2}\Big(\widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{y}),\ \boldsymbol{0},\ -1\Big) = \Big(\boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{y}),\ \boldsymbol{0}\Big),
\end{aligned}$$

implying

$$\Big[\boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\boldsymbol{y},\ \boldsymbol{0},\ 2r_1+1,\ 2r_1+3\Big)\Big]_{[1:d_2]} = \boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{y}).$$

Define $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d+2}$ via

$$\boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) \coloneqq \Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}),\ \boldsymbol{0},\ 2r_1+1,\ 2r_1+3\Big) \in \mathbb{R}^{d+2} \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^{d_0}$$

and $\boldsymbol{\mathcal{L}}_2 : \mathbb{R}^{d+2} \to \mathbb{R}^{d_3}$ via

$$\boldsymbol{\mathcal{L}}_2(\boldsymbol{u}, \boldsymbol{v}) \coloneqq \widetilde{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{u}) \quad \text{for any } \boldsymbol{u} \in \mathbb{R}^{d_2} \text{ and } \boldsymbol{v} \in \mathbb{R}^{d+2-d_2}.$$

Then, for any $\boldsymbol{x} \in [-A, A]^{d_0}$, we have $\boldsymbol{y} = \widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) \in [-\widetilde{A}, \widetilde{A}]^{d_1}$, implying

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}^{\circ(r_1+r_2+1)} \circ \boldsymbol{\mathcal{L}}_1(\boldsymbol{x}) &= \boldsymbol{\mathcal{L}}_2 \circ \boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}),\ \mathbf{0},\ 2r_1+1,\ 2r_1+3\Big) \\
&= \boldsymbol{\mathcal{L}}_2\Big(\boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\boldsymbol{y},\ \mathbf{0},\ 2r_1+1,\ 2r_1+3\Big)\Big) \\
&= \widetilde{\boldsymbol{\mathcal{L}}}_3\Big(\Big[\boldsymbol{g}^{\circ(r_1+r_2+1)}\Big(\boldsymbol{y},\ \mathbf{0},\ 2r_1+1,\ 2r_1+3\Big)\Big]_{[1:d_2]}\Big) \\
&= \widetilde{\boldsymbol{\mathcal{L}}}_3\Big(\boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{y})\Big) = \widetilde{\boldsymbol{\mathcal{L}}}_3 \circ \boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}).
\end{aligned}$$

So we finish the proof of Proposition 3.3. □

E.2. Proof of Lemma E.1

The proof of Lemma E.1 will be provided after establishing an auxiliary lemma, namely Lemma E.2 below. As we shall see later, the auxiliary lemma plays a crucial role in the proof of Lemma E.1.

Lemma E.2. *For any $M > 0$ and $d \in \mathbb{N}^+$, there exists $\boldsymbol{\phi} \in \mathcal{NN}\{2d+2,\ 1;\ \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}\}$ such that*

$$\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}, t) = (\boldsymbol{z}, t) \quad \text{with} \quad \boldsymbol{z} = \begin{cases} \boldsymbol{x} & \text{if } t \geq 1 \\ \boldsymbol{y} & \text{if } t \leq -1 \end{cases}$$

for any $\boldsymbol{x}, \boldsymbol{y} \in [-M, M]^d$ and $t \in (-\infty, -1] \cup [1, \infty)$.

Proof. The key idea for proving this lemma is to use a ReLU network to realize a selector function $g : \mathbb{R}^3 \to \mathbb{R}$ such that

$$g(u, v, t) = \begin{cases} u & \text{if } t \geq 1 \\ v & \text{if } t \leq -1 \end{cases}$$

for any $u, v \in [-M, M]$ and $t \in (-\infty, -1] \cup [1, \infty)$. To this end, we define

$$g(u, v, t) \coloneqq \sigma\big(u + Mt\big) + \sigma\big(v - Mt\big) - M\sigma(t) - M\sigma(-t). \tag{19}$$

Let us verify that g meets the requirements.

In the case of $t \geq 1$, we have $u + Mt \geq 0$ and $v - Mt \leq 0$ for any $u, v \in [-M, M]$, implying

$$\begin{aligned}
g(u, v, t) &= \sigma\big(u + Mt\big) + \sigma\big(v - Mt\big) - M\sigma(t) - M\sigma(-t) \\
&= (u + Mt) + 0 - Mt - 0 = u.
\end{aligned}$$

In the case of $t \leq -1$, we have $u + Mt \leq 0$ and $v - Mt \geq 0$ for any $u, v \in [-M, M]$, implying

$$\begin{aligned}
g(u, v, t) &= \sigma\big(u + Mt\big) + \sigma\big(v - Mt\big) - M\sigma(t) - M\sigma(-t) \\
&= 0 + (v - Mt) - 0 - M \cdot (-t) = v.
\end{aligned}$$

Based on g, we can design a ReLU network to realize $\boldsymbol{\phi} : \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}$ that maps $(\boldsymbol{x}, \boldsymbol{y}, t)$ to

$$(\boldsymbol{z}, t) = (z_1, \cdots, z_d, t) = \Big(g(x_1, y_1, t),\ \cdots,\ g(x_d, y_d, t), \quad t\Big)$$

for any $\boldsymbol{x} = (x_1, \cdots, x_d), \boldsymbol{y} = (y_1, \cdots, y_d) \in [-M, M]^d$ and $t \in (-\infty, -1] \cup [1, \infty)$.

We present the ReLU network realizing $\boldsymbol{\phi}$ in Figure 15. Clearly, $\boldsymbol{\phi} \in \mathcal{NN}\{2d+2,\ 1;\ \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}\}$. So we finish the proof of Lemma E.2. □

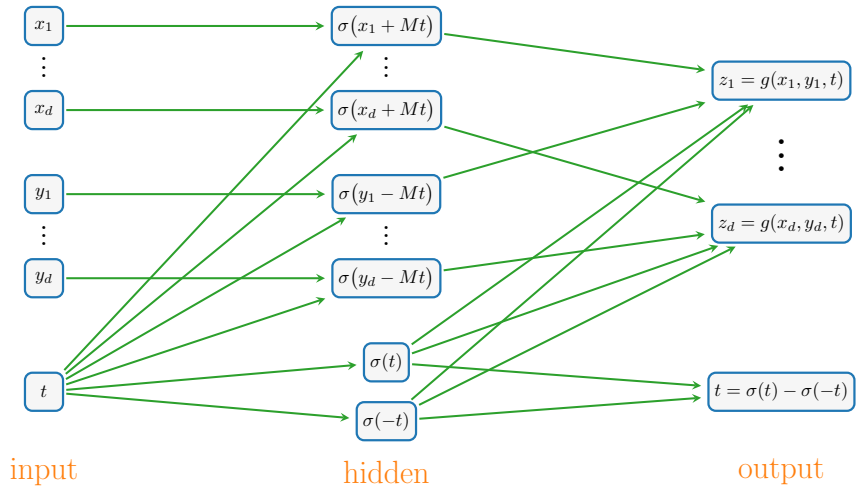


Figure 15. An illustration of the target ReLU network realizing ϕ based on Equation (19).

Equipped with Lemma E.2, we are prepared to demonstrate the proof of Lemma E.1.

Proof of Lemma E.1. We will construct $\boldsymbol{\Phi} = \boldsymbol{\psi} \circ \boldsymbol{G}$ via two steps below.

- First, we construct $\boldsymbol{G} : \mathbb{R}^{d+1} \to \mathbb{R}^{2d+1}$ by stacking $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, and g_0, where $g_0(t) = t - 2$.
- Next, we will apply Lemma E.2 to construct a selector function $\boldsymbol{\psi} : \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}$, determining which sub-block ($\boldsymbol{g}_1$ or $\boldsymbol{g}_2$) in $\boldsymbol{G}$ is used in each composition.

More details can be found below.

Step 1: Constructing $\boldsymbol{G}$.

Recall the ℓ^∞-norm of a vector $\boldsymbol{a} = (a_1, a_2, \cdots, a_d) \in \mathbb{R}^d$ is given by

$$\|\boldsymbol{a}\|_{\ell^\infty} = \|\boldsymbol{a}\|_\infty \coloneqq \max\big\{|a_i| : i = 1, 2, \cdots, d\big\}.$$

Set

$$M = \max\{M_k : k = 0, 1, \cdots, r_1 + r_2\},$$

where $M_0 = \max\{A,\ 100(r_1 + r_2 + 1)\}$ and M_k is given by

$$M_k = \sup\Big\{\big\|\boldsymbol{h}_k \circ \cdots \circ \boldsymbol{h}_1(\boldsymbol{x})\big\|_{\ell^\infty} : \boldsymbol{x} \in [-A, A]^d, \quad \boldsymbol{h}_1, \cdots, \boldsymbol{h}_k \in \{\boldsymbol{g}_1, \boldsymbol{g}_2\}\Big\}$$

for $k = 1, 2, \cdots, r_1 + r_2$.

Define $\boldsymbol{G} : [-M, M]^{d+1} \to [-M, M]^{2d+1}$ via

$$\boldsymbol{G}(\boldsymbol{x}, t) \coloneqq \Big(\boldsymbol{g}_1(\boldsymbol{x}),\ \boldsymbol{g}_2(\boldsymbol{x}),\ g_0(t)\Big) \quad \text{for any } (\boldsymbol{x}, t) \in [-M, M]^{d+1},$$

where $g_0(t) = t - 2$. Recall that $\boldsymbol{g}_i \in \mathcal{NN}\big\{N_i,\ L_i;\ \mathbb{R}^d \to \mathbb{R}^d\big\}$ for $i = 1, 2$. To make $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, and g_0 have the same number of hidden layers, we need to manually add some "trifling" layers.

Then, by setting $L = \max\{L_1, L_2\}$ and

we have

$$\boldsymbol{g}_1(\boldsymbol{x}) = \sigma^{\circ(L - L_1)} \circ (\boldsymbol{g}_1 + M)(\boldsymbol{x}) - M,$$

$$\boldsymbol{g}_2(\boldsymbol{x}) = \sigma^{\circ(L-L_2)} \circ (\boldsymbol{g}_2 + M)(\boldsymbol{x}) - M,$$

and

$$g_0(t) = \sigma^{\circ L} \circ g_0(t + M + 2) - M - 2$$

for any $(\boldsymbol{x}, t) \in [-M, M]^{d+1}$. Then, $\boldsymbol{g}_i \in \mathcal{NN}\big\{ \max\{N_i, d\},\ L;\ \mathbb{R}^d \to \mathbb{R}^d \big\}$ for $i = 1, 2$ and $g_0 \in \mathcal{NN}\big\{1,\ L;\ \mathbb{R} \to \mathbb{R}\big\}$. It follows that

$$\boldsymbol{G} \in \mathcal{NN}\Big\{ \max\{N_1, d\} + \max\{N_2, d\} + 1,\ L = \max\{L_1, L_2\};\ \mathbb{R}^{d+1} \to \mathbb{R}^{2d+1} \Big\}.$$

Step 2: Constructing $\boldsymbol{\psi}$.

Next, let construct a selector function $\boldsymbol{\psi}$ to "select" $\boldsymbol{g}_1$ or $\boldsymbol{g}_2$ in $\boldsymbol{G}$.

By Lemma E.2, there exists

$$\boldsymbol{\psi} \in \mathcal{NN}\{2d + 2,\ 1;\ \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}\}$$

such that

$$\boldsymbol{\psi}(\boldsymbol{u}, \boldsymbol{v}, t) = (\boldsymbol{w}, t) \quad \text{with} \quad \boldsymbol{w} = \begin{cases} \boldsymbol{u} & \text{if } t \geq 1 \\ \boldsymbol{v} & \text{if } t \leq -1 \end{cases} \tag{20}$$

for any $\boldsymbol{u}, \boldsymbol{v} \in [-M, M]^d$ and $t \in (-\infty, -1] \cup [1, \infty)$. Then, we can define the desired $\boldsymbol{\Phi}$ via $\boldsymbol{\Phi} \coloneqq \boldsymbol{\psi} \circ \boldsymbol{G}$. Clearly, $\boldsymbol{G} \in \mathcal{NN}\big\{ \max\{N_1, d\} + \max\{N_2, d\} + 1,\ \max\{L_1, L_2\};\ \mathbb{R}^{d+1} \to \mathbb{R}^{2d+1} \big\}$ and $\boldsymbol{\psi} \in \mathcal{NN}\{2d + 2,\ 1;\ \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}\}$ imply

$$\begin{aligned} \boldsymbol{\Phi} = \boldsymbol{\psi} \circ \boldsymbol{G} &\in \mathcal{NN}\Big\{ \max\Big\{ \max\{N_1, d\} + \max\{N_2, d\} + 1,\ 2d + 2 \Big\},\ \max\{L_1, L_2\} + 1;\ \mathbb{R}^{d+1} \to \mathbb{R}^{d+1} \Big\} \\ &\subseteq \mathcal{NN}\Big\{ N_1 + N_2 + 2d,\ \max\{L_1, L_2\} + 1;\ \mathbb{R}^{d+1} \to \mathbb{R}^{d+1} \Big\}. \end{aligned}$$

It remains to verify that

$$\boldsymbol{\Phi}^{\circ(r_1+r_2)}(\boldsymbol{x},\ 2r_1 + 1) = \Big(\boldsymbol{g}_2^{\circ r_2} \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x}),\ -2r_2 + 1 \Big) \quad \text{for any } \boldsymbol{x} \in [-A, A]^d.$$

Fix $\boldsymbol{x} \in [-A, A]^d$ and we can write

$$\big(\boldsymbol{\xi}_k,\ t_k\big) = \boldsymbol{\Phi}^{\circ k}(\boldsymbol{x},\ 2r_1 + 1) \quad \text{with } \boldsymbol{\xi}_k \in \mathbb{R}^d \quad \text{for } k = 0, 1, \cdots, r_1 + r_2,$$

where $\boldsymbol{\Phi}^{\circ 0}$ means the identity map.

Observe that, for $k = 1, 2, \cdots, r_1 + r_2$,

$$\begin{aligned} \big(\boldsymbol{\xi}_k,\ t_k\big) &= \boldsymbol{\Phi}^{\circ k}(\boldsymbol{x},\ 2r_1 + 1) = \boldsymbol{\Phi} \circ \boldsymbol{\Phi}^{\circ(k-1)}(\boldsymbol{x},\ 2r_1 + 1) \\ &= \boldsymbol{\Phi}(\boldsymbol{\xi}_{k-1}, t_{k-1}) = \boldsymbol{\psi} \circ \boldsymbol{G}(\boldsymbol{\xi}_{k-1}, t_{k-1}) = \boldsymbol{\psi}\Big(\boldsymbol{g}_1(\boldsymbol{\xi}_{k-1}),\ \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1}),\ g_0(t_{k-1}) \Big). \end{aligned}$$

Then, by Equation (20), it is easy to verify that $t_k = g_0(t_{k-1}) = t_{k-1} - 2$ and $\boldsymbol{g}_1(\boldsymbol{\xi}_{k-1}),\ \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1}) \in [-M, M]^d$ for $k = 1, 2, \cdots, r_1 + r_2$, from which we deduce

$$t_k = t_0 - 2k = 2r_1 + 1 - 2k = 2(r_1 - k) + 1$$

and

$$\boldsymbol{\xi}_k = \begin{cases} \boldsymbol{g}_1(\boldsymbol{\xi}_{k-1}) & \text{if } t_k = g_0(t_{k-1}) \geq 1 \\ \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1}) & \text{if } t_k = g_0(t_{k-1}) \leq -1 \end{cases}$$

for $k = 1, 2, \cdots, r_1 + r_2$.

Moreover, for $k = 1, 2, \cdots, r_1$, we have $t_k = 2(r_1 - k) + 1 \geq 1$ and hence $\boldsymbol{\xi}_k = \boldsymbol{g}_1(\boldsymbol{\xi}_{k-1})$, implying

$$\boldsymbol{\xi}_{r_1} = \boldsymbol{g}_1(\boldsymbol{\xi}_{r_1-1}) = \cdots = \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{\xi}_0) = \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x}).$$

For $k = r_1 + 1, r_1 + 2, \cdots, r_1 + r_2$, we have $t_k = 2(r_1 - k) + 1 \leq -1$ and hence $\boldsymbol{\xi}_k = \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1})$, implying

$$\boldsymbol{\xi}_{r_1+r_2} = \boldsymbol{g}_2(\boldsymbol{\xi}_{r_1+r_2-1}) = \cdots = \boldsymbol{g}_2^{\circ r_2}(\boldsymbol{\xi}_{r_1}) = \boldsymbol{g}_2^{\circ r_2}\big(\boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x})\big) = \boldsymbol{g}_2^{\circ r_2} \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x}).$$

Therefore, we have

$$\Big[\boldsymbol{\Phi}^{\circ(r_1+r_2)}(\boldsymbol{x},\, 2r_1 + 1)\Big]_{[1:d]} = \boldsymbol{g}_2^{\circ r_2} \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x})$$

and

$$\Big[\boldsymbol{\Phi}^{\circ(r_1+r_2)}(\boldsymbol{x},\, 2r_1 + 1)\Big]_{[d+1]} = t_{r_1+r_2} = \big(r_1 - (r_1 + r_2) + 1\big) = -2r_2 + 1,$$

from which we deduce

$$\boldsymbol{\Phi}^{\circ(r_1+r_2)}(\boldsymbol{x},\, 2r_1 + 1) = \Big(\boldsymbol{g}_2^{\circ r_2} \circ \boldsymbol{g}_1^{\circ r_1}(\boldsymbol{x}),\, -2r_2 + 1\Big).$$

Thus, we complete the proof of Lemma E.1. □