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

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

Abstract

This paper studies the expressive power of deep neural networks from the perspective of function compositions. We show that repeated compositions of a single fixed-size ReLU network can produce super expressive power. In particular, 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 $\boldsymbol{g}^{\circ r}$ means 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 good approximation power even if its dynamics function is time-independent and realized by a fixed-size ReLU network.

1. Introduction

The study of expressive power for deep neural networks has become an active topic in recent years, motivated by the tremendous success of deep neural networks in various learning tasks. The success of deep learning is generally accompanied by explosive growth in the size of deep learning models, which makes the training challenging and computationally expensive. To bridge this gap, numerous model compression and acceleration methods have recently been proposed to handle computationally expensive models. The key idea of most of these methods is to share parameters. Parameter-sharing schemes used in neural networks can control the overall number of parameters for reducing memory and computation costs. This paper explores this topic from the perspective of function compositions. We study the expressive power of a particular network architecture generated via repeated compositions of a single fixed-size network. In our new network architecture, we share parameters via repetitions of a single network. We will focus on the ReLU (rectified linear unit) activation function and use it to demonstrate our ideas. We prove that repeated compositions of a single fixed-size ReLU network can produce super expressive power, though itself only has pretty weak expressive power.

To simplify the notation, we use $\mathcal{NN}\{N,\ L;\ \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}\}$ to denote the function set consisting of any function $\boldsymbol{\phi} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ realized by a ReLU network of width $N \in \mathbb{N}^+$ and depth $L \in \mathbb{N}^+$. In our setting, the width of a network means the maximum number of neurons in a hidden layer and the depth means 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, $\boldsymbol{g}^{\circ 0}$ means 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 exists $\boldsymbol{g} \in \mathcal{NN}\{69d+48,\ 5;\ \mathbb{R}^{5d+5} \to \mathbb{R}^{5d+5}\}$ 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*

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

We remark that $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ in Theorem 1.1 are used to match the dimensions and hence can be replaced by many other functions with the desired input and output dimensions. Here, we adopt a simple choice by letting them be affine. In Theorem 1.1, we introduce a new network architecture designed via repeated compositions of a single subnetwork, which will be referred as repeated-composition networks (RCNets). The hypothesis space of the RCNet corresponding to $\boldsymbol{g}$ is defined as

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

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

Then we have an immediate corollary as follows.

Corollary 1.2. *Given any $p \in [1,\infty)$, suppose $\mathcal{H}(\boldsymbol{g})$ is defined just above and set $\mathcal{G} = \mathcal{NN}\{69d+48,\, 5;\, \mathbb{R}^{5d+5} \to \mathbb{R}^{5d+5}\}$. Then $\mathcal{H} = \cup_{\boldsymbol{g}\in\mathcal{G}}\mathcal{H}(\boldsymbol{g})$ 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 $p \in [1,\infty)$. Hence, $\mathcal{H}$ is dense in $L^p([0,1]^d)$ in terms of the L^p-norm for any $p \in [1,\infty)$.We remark that $\mathcal{G}$ in Corollary 1.2 is generated by a fixed-size ReLU network and hence is a set of continuous piecewise linear functions with (at most) a fixed number of pieces.

Note that the approximation error in Theorem 1.1 is characterized by the L^p-norm for any $p \in [1,\infty)$. We can generalize it to the L^∞-norm, though the constants are much larger.

Theorem 1.3. *Given any $f \in C([0,1]^d)$ and $r \in \mathbb{N}^+$, there exists $\boldsymbol{g} \in \mathcal{NN}\{4^{d+5}d,\, 3+2d;\, \mathbb{R}^{\widetilde{d}} \to \mathbb{R}^{\widetilde{d}}\}$ 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*

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

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

The main ideas of 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 case of $\alpha = 1$, i.e., f is a Lipschitz continuous function with a Lipschitz constant $\lambda > 0$, then 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. The results in (Shen et al., 2020; Yarotsky, 2018; Zhang, 2020) imply that the optimal approximation error is $\mathcal{O}(r^{-2/d})$ 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 $\mathcal{O}(r^{-1/d})$ when $\boldsymbol{g}$ 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 share 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. Moreover, our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ is expected to have better gradient behavior since the gradient with respect to the parameters in $\boldsymbol{g}$ would not vanish for large r.

Next, we point out some relation between our approximation results and dynamical systems. Our results reveal that a continuous-depth network generated via a dynamical system has good approximation power even if the dynamics is time-independent (time-invariant) and realized by a fixed-size ReLU network.

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

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

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 $\boldsymbol{F}$.

Define $\boldsymbol{g}_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}^n$ via

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

and suppose $\boldsymbol{F}(\boldsymbol{y}, t, \boldsymbol{\theta})$ is independent of t for any $(\boldsymbol{y}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \boldsymbol{\Theta}$. 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 good 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 good 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 good 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 rest of this paper is organized as follows. We connect our results to existing work in Section 2. In Section 3, we present the main ideas of proving Theorems 1.1 and 1.3. Next, we use two simple experiments to numerically show the super-approximation power of repeated compositions of a single fixed-size ReLU network. Finally, Section 5 concludes this paper with a short discussion.

2. Related Work

In this section, we will summarize previous research related to our results. We first point out the connection between deep learning and dynamical systems. Next, we discuss the parameter-sharing schemes of neural networks. Finally, we compare our results with existing ones from an approximation perspective.

2.1. Deep learning via dynamical systems

A dynamical system describes the evolution of something over time, the concept of which has its origins in Newtonian mechanics. One may refer to (Holmes, 2007) for a short summary of the history of dynamical systems. Generally, a dynamical system has two key components as follows. One is the state variable(s), which can completely describe the state of the dynamical system. The other one is the time evolution rule, which describes what future states follow from the current state.

As mentioned previously, a dynamical system can be described by an ODE in Equation (1). A novel idea is proposed in (E, 2017) to understand the discretization of a continuous dynamical system as a continuous-depth residual network. The author of (E, 2017) connects dynamical systems to deep learning by discussing the idea of using continuous dynamical systems to model general high-dimensional nonlinear functions used in machine learning.

Recently, there are a considerable amount of publications connecting dynamical systems to deep learning. The authors of (Chen et al., 2018) parameterize the derivative of the hidden state using a neural network and introduce continuous-depth residual networks. Several advantages of continuous-depth models (e.g., constant memory cost) are demonstrated therein. The authors of (Li et al., to appear) connecting dynamical systems approach to deep learning by establishing general sufficient conditions for the universal approximation property of continuous-depth residual networks, which are designed based on dynamical systems. Later in (Li et al., 2022), they extend their results from generic continuous functions to specific invariant functions. The universal approximation property of deep fully convolutional networks is studied in (Lin et al., 2022) from the perspective of dynamical systems. It is shown therein that deep residual fully convolutional networks and their continuous-depth counterpart of constant channel width can achieve universal approximation of specific symmetric functions.

2.2. Discussion from a parameter-sharing perspective

In recent years, depth in neural networks has shown remarkable success, which generally results in computationally expensive models. Numerous model compression and acceleration methods have recently been proposed to reduce computation costs. The idea of parameter sharing is the core of most of these methods. Parameter-sharing schemes are used in neural networks to control the overall number of parameters for reducing memory and computation costs. Our network architecture, generated via repeated compositions of a single fixed-size network, can be regarded as a specific parameter sharing scheme of neural networks.

To the best of our knowledge, parameter-sharing schemes of neural networks can be roughly divided into three basic cases. The first one is to share parameters in the same layer. A typical network example with this scheme is the convolutional neural network (CNN). In convolutional operations of CNN architectures, kernels (filters) are shared across all image positions. The second scheme is to share parameters

among different layers of neural networks, e.g., recurrent neural networks. This scheme is used in our network architecture. To be exact, we share parameters via compositions of a single fixed-size network and show that repetitions of a fixed number of parameters can produce super approximation power. Beyond the above two parameter sharing schemes, one may also share parameters among different neural networks or models, especially for multi-task learning. For a further discussion on parameter sharing in neural networks, one may refer to (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 deep neural networks has been widely studied in the past decades. Numerous publications focus on constructing all kinds of neural networks to approximate various target functions, e.g., (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., 2019; Li et al., to appear; Montanelli & Yang, 2020; Shen et al., 2019; 2020; Lu et al., 2021; Zhang, 2020; Shen et al., 2022b;a). In the early stage of such a topic, the universal approximation power of one-hidden-layer networks is explored. The universal approximation theorem (Cybenko, 1989; Hornik, 1991; Hornik et al., 1989) showed that a sufficiently large neural network can approximate a certain type of target function arbitrarily well, without estimating the approximation error in terms of the network size.

Recently, a large number of publications concentrate on the approximation of deep neural networks. The results in (Shen et al., 2020; Yarotsky, 2018; Zhang, 2020) imply the optimal approximation error $\mathcal{O}(n^{-2/d})$ while using ReLU networks with at most n parameters to approximate 1-Lipschitz continuous functions on $[0,1]^d$. This optimal approximation error can be further improved by introducing additional assumptions. There are generally two main ideas to get better approximation errors. The first one is to consider smaller function spaces, e.g., smooth functions (Lu et al., 2021; Yarotsky & Zhevnerchuk, 2020) and band-limited functions (Montanelli et al., 2021). The other one is to design new networks, e.g., Floor-ReLU networks (Shen et al., 2021a), Floor-Exponential-Step (FLES) networks (Shen et al., 2021b), (Sin, ReLU, 2^x)-activated networks (Jiao et al., 2021), three-dimensional networks (Shen et al., 2022) by introducing additional dimension called height beyond width and depth. The literature on approximation analysis of neural networks is vast and we just name a few due to the page limit.

This paper proposes a specific neural network architecture generated by repeated compositions of a single fixed-size network. As shown in Theorems 1.1 and 1.3, repetitions of one small ReLU network block can produce super approximation power. We will conduct experiments to numerically verify our theoretical results in Section 4.

3. Ideas of proving Theorems 1.1 and 1.3

Let us present the key ideas of proving Theorems 1.1 and 1.3. We hope to construct a piecewise constant function to approximate the target continuous function. However, the continuity of ReLU networks makes them unable to uniformly approximate piecewise constant functions well. Thus, we first ignore the approximation in a small region and design ReLU networks to realize piecewise constant functions outside this small region to approximate the target function. To this end, 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 $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. In this step, we need to show 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.

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 $\boldsymbol{\Phi}_1$ 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

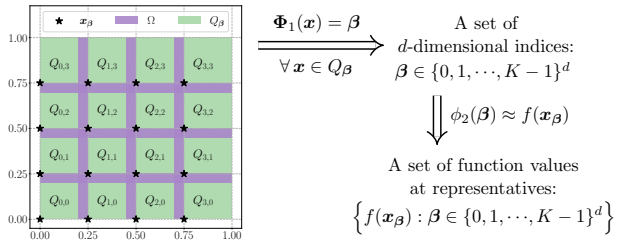


Figure 1. An illustration of the ideas of constructing the desired function $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$. Note that $\phi \approx f$ outside Ω since $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})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

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 $n, m \in \mathbb{N}^+$ with $n \leq m$, there exists $\boldsymbol{g} \in \mathcal{NN}\{9,\, 1;\, \mathbb{R}^5 \to \mathbb{R}^5\}$ 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*

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \boldsymbol{\mathcal{L}}_1(x) = k \tag{3}$$

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

Step 2: Constructing ϕ_2

The goal of ϕ_2 is to map $\boldsymbol{\beta}$ approximation to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. In other words, we need to construct ϕ_2 to fit K^d points $\big(\boldsymbol{\beta}, f(\boldsymbol{x}_{\boldsymbol{\beta}})\big) \in \mathbb{R}^{d+1}$ for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. We remark that, 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 $\widetilde{\phi}_2$ to fit $\big(\mathcal{L}(\boldsymbol{\beta}), f(\boldsymbol{x}_{\boldsymbol{\beta}})\big) \in \mathbb{R}^2$ for $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$ since $\phi_2 = \widetilde{\phi}_2 \circ \mathcal{L}$ can map $\boldsymbol{\beta}$ approximate 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, a proposition is introduced to simplify the construction of $\widetilde{\phi}_2$.

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

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

Then, there exists $\boldsymbol{g} \in \mathcal{NN}\{16,\, 2;\, \mathbb{R}^6 \to \mathbb{R}^6\}$ 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

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

The proof of Proposition 3.2 is complicated and hence is placed in Section D of the appendix.

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 $\phi := \phi_2 \circ \boldsymbol{\Phi}_1$ 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 pointwisely 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:

$$\phi = \widetilde{\boldsymbol{\mathcal{L}}}_3 \circ \boldsymbol{g}_2^{\circ r_2} \circ \widetilde{\boldsymbol{\mathcal{L}}}_2 \circ \boldsymbol{g}_1^{\circ r_1} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1, \tag{4}$$

where $\widetilde{\boldsymbol{\mathcal{L}}}_1, \widetilde{\boldsymbol{\mathcal{L}}}_2, \widetilde{\boldsymbol{\mathcal{L}}}_3$ 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 (4) to the desired one. A proposition is established to make such a conversion.

Proposition 3.3. *Given there affine linear maps $\widetilde{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$, $\widetilde{\boldsymbol{\mathcal{L}}}_2 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, and $\widetilde{\boldsymbol{\mathcal{L}}}_3 : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$, 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 exists $\boldsymbol{g} \in \mathcal{NN}\{N_1 + N_2 + 6d + 2,\, \max\{L_1 + 2, L_2 + 1\};\, \mathbb{R}^{d+2} \to \mathbb{R}^{d+2}\}$ 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

In this section, we will conduct experiments as a proof of concept to explore the numerical properties of our RCNet

architecture. As shown in Theorems 1.1 and 1.3, our RCNet architecture can produce super approximation power via repeated compositions of a single fixed-size ReLU network. The goal of our experiments is to numerically verify our theoretical results. To be exact, we hope to show increasing r in our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ could improve the approximation results, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are affine linear maps and $\boldsymbol{g}$ is a ReLU network block. To this end, we conduct two different experiments: one for function approximation in Section 4.1 and the other one for classification in Section 4.2.

Next, let us roughly discuss when better experiment results can be expected by increasing r in our RCNet architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$. As we know, the test error is includes three errors: approximation, generalization, and optimization errors. In our discussion, we suppose the optimization error is well-controlled since our experiments are simple. Clearly, increasing r can reduce the approximation error, but may result in a large generalization error. Thus, we would expect that a larger r could perform better if the approximation error is the leading term. That means, we can either make the approximation error relatively large or make the generalization error small. To make the approximation error relatively large, we choose a sufficiently complicated target function in the first experiment and adopt a difficult binary classification problem in the second experiment. In both experiments, sufficiently many samples are generated to control the generalization error.

4.1. Function approximation

We will use our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ to solve a function approximation problem, where $\boldsymbol{\mathcal{L}}_1$ and $\mathcal{L}_2$ are two affine linear maps and $\boldsymbol{g}$ is (realized by) a ReLU network block. To better compare the approximation power of our network architecture for different r, we choose a complicated (oscillatory) function f as the target function, where $f : [0,1]^2 \to \mathbb{R}$ is defined via

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

for any $\boldsymbol{x} = (x_1, x_2) \in [0,1]^2$, where

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

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

See an illustration of f in Figure 2.

In our experiment, we let $\boldsymbol{g}$ be a ReLU network block with 2 hidden layers and each hidden layer has 100 neurons, i.e., $\boldsymbol{g} \in \mathcal{NN}\{100,\, 2;\, \mathbb{R}^{100} \to \mathbb{R}^{100}\}$. In other words, $\boldsymbol{g}$ can be

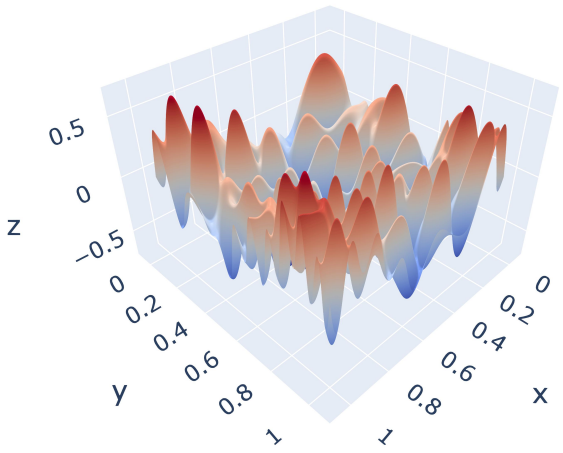


Figure 2. An illustration of the target function f.

defined via

$$\boldsymbol{g}(\boldsymbol{x}) := \boldsymbol{A}_2 \sigma\Big(\boldsymbol{A}_1 \sigma(\boldsymbol{A}_0 \boldsymbol{x} + \boldsymbol{b}_0) + \boldsymbol{b}_1\Big) + \boldsymbol{b}_2$$

for any $\boldsymbol{x} \in \mathbb{R}^{100}$, where $\boldsymbol{A}_i \in \mathbb{R}^{100\times 100}$ and $\boldsymbol{b}_i \in \mathbb{R}^{100}$ are parameters for $i = 0, 1, 2$ and σ is the ReLU activation function that can be applied to a vector entry-wisely. Then, the input and output dimensions of two affine linear maps $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^2 \to \mathbb{R}^{100}$ and $\mathcal{L}_2 : \mathbb{R}^{100} \to \mathbb{R}$ are correspondingly given. We will use $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ to approximate the target function f for $r = 1, 2, 3, 4$ and we aim to numerically show that increasing r would improve the approximation error. We remark that there are $(100^2 + 100) \times 3 = 30300$ parameters in $\boldsymbol{g}$, $2\times 100 + 100 = 300$ parameters in $\boldsymbol{\mathcal{L}}_1$, and $100 \times 1 + 1 = 101$ parameters in $\mathcal{L}_2$. Thus, the total number of parameters in $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ is $30300+300+101 = 30701$ for $i = 1, 2, 3, 4$. For $r \geq 2$, parameters of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ are partially shared via repetitions of $\boldsymbol{g}$.

Before presenting the numerical results, let us first present the hyper-parameters for training our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$. We use the uniform distribution to generate 10^6 training samples and 10^5 test samples in $[0,1]^2$. We adopt RAdam (Liu et al., 2020) as the optimization method. The batch size, the number of epochs, and the learning rate in epochs $5(i-1)+1$ to $5i$ for $i = 1, 2, \cdots, 200$ are set to 512, 1000, and $0.002 \times 0.9^{i-1}$, respectively. To better estimate the training and test losses, we adopt three loss functions: the mean squared error (MSE), the mean absolute error (MAE), and the maximum (MAX) loss functions. Let $\phi = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1$ be the network-generated function and $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$ be the training (test) samples. Then, the MSE, MAE, and MAX losses are given by $\frac{1}{m}\sum_{i=1}^{m} \big(\phi(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\big)^2$, $\frac{1}{m}\sum_{i=1}^{m} \big|\phi(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\big|$, and $\max\big\{|\phi(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)| : i = 1, 2, \cdots, m\big\}$, respectively. The MSE loss is used to train our model. To make the experiment more reliable, we repeat our experiment 12 times

and discard 2 top-performing and 2 bottom-performing trials by measuring the performance via the average test loss (MSE) of the last 100 epochs. For each epoch, the average of training (test) losses in the rest 8 trials is adopted as the target training (test) loss.

Now, we are ready to present the experiment results to compare the numerical performances of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ for $r = 1, 2, 3, 4$. Training and test losses (MSE) over epochs for $r = 1, 2, 3, 4$ are summarized in Figure 3. To better compare the order of training and test losses, we plot epochs and the base 10 logarithms of losses as x-coordinates and y-coordinates, respectively.

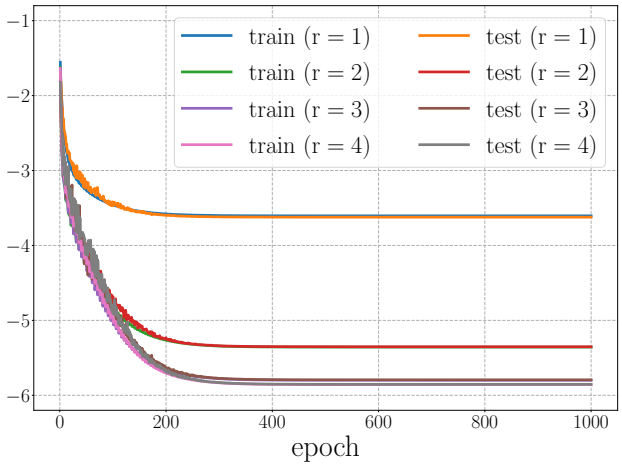


Figure 3. The base 10 logarithms of training and test losses over epochs.

In Table 1, we compare the numerical results of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ for $r = 1, 2, 3, 4$ via the average test losses of the last 100 epochs measured in several loss functions. As we can see from Figure 3 and Table 1, increasing r improves the training and test losses significantly as expected. The experiment results numerically verify our theoretical ones shown in Theorems 1.1 and 1.3.

Table 1. Test loss comparison.

$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$	test loss		
	MSE	MAE	MAX
$r = 1$	2.39×10^{-4}	1.09×10^{-2}	1.38×10^{-1}
$r = 2$	4.45×10^{-6}	1.59×10^{-3}	1.52×10^{-2}
$r = 3$	1.61×10^{-6}	9.73×10^{-4}	8.19×10^{-3}
$r = 4$	1.40×10^{-6}	9.12×10^{-4}	7.85×10^{-3}

4.2. Classification

We adopt the binary classification experiment proposed in (Shen et al., 2022) based on the Archimedean spiral. The goal of such a classification problem is to classify samples in two disjoint sets $\mathcal{S}_0$ and $\mathcal{S}_1$ (see Figure 4 for an illustration), which are designed based on the Archimedean spiral.

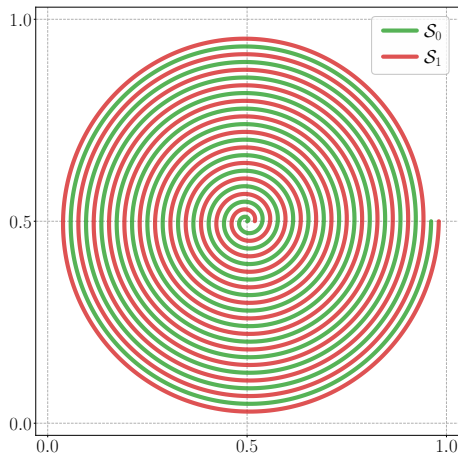


Figure 4. An illustration for $\mathcal{S}_0$ and $\mathcal{S}_1$.

Let us present the details of constructing $\mathcal{S}_0$ and $\mathcal{S}_1$. As we know, an Archimedean spiral can be described by the equation $r = a + b\theta$ in polar coordinates (r, θ) for proper $a, b \in \mathbb{R}$. Let us first define two curves

$$\widetilde{\mathcal{C}}_i := \left\{ (x, y) : \begin{cases} x = r_i \cos\theta, \\ y = r_i \sin\theta, \end{cases} r_i = a_i + b_i\theta,\ \theta \in [0, s\pi] \right\}$$

for $i = 0, 1$, where $a_0 = 0$, $a_1 = 1$, $b_0 = b_1 = 1/\pi$, and $s = 24$. Next, we normalize $\widetilde{\mathcal{C}}_i$ as $\mathcal{C}_i \subseteq [0, 1]^2$ for each $i \in \{0, 1\}$, where $\mathcal{C}_i$ is defined as

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

for $i = 0, 1$. With $\mathcal{C}_0$ and $\mathcal{C}_1$ in hand, we can define the two desired sets $\mathcal{S}_0$ and $\mathcal{S}_1$ via

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

for $i = 0, 1$, where $\varepsilon = 0.006$ in our experiments. See an illustration for $\mathcal{S}_0$ and $\mathcal{S}_1$ in Figure 4.

We will use our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ to classify samples in $\mathcal{S}_0 \cup \mathcal{S}_1$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are two affine linear maps and $\boldsymbol{g}$ is (realized by) a ReLU network block. In this experiment, $\boldsymbol{g}$ is generated via the composition of an affine linear map and the ReLU activation function, i.e., $\boldsymbol{g}$ can be defined via

$$\boldsymbol{g}(\boldsymbol{x}) := \sigma(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$$

for any $\boldsymbol{x} \in \mathbb{R}^{72}$, where $\boldsymbol{A} \in \mathbb{R}^{72 \times 72}$ and $\boldsymbol{b} \in \mathbb{R}^{72}$ are parameters and σ is the ReLU activation function that can

be applied to a vector entry-wisely. Then, the input and output dimensions of two affine linear maps $\mathcal{L}_1 : \mathbb{R}^2 \to \mathbb{R}^{72}$ and $\mathcal{L}_2 : \mathbb{R}^{72} \to \mathbb{R}^2$ are correspondingly given. We will use $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ to classify samples in $\mathcal{S}_0 \cup \mathcal{S}_1$ for $r = 1, 2, 3, 4$ and our goal is to numerically show that increasing r would improve the test accuracy. We remark that there are $72^2 + 72 = 5256$ parameters in $\boldsymbol{g}$, $2 \times 72 + 72 = 216$ parameters in $\mathcal{L}_1$, and $72 \times 2 + 2 = 146$ parameters in $\mathcal{L}_2$. Thus, the total number of parameters in $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ is $5256 + 216 + 146 = 5618$ for $i = 1, 2, 3, 4$. For $r \geq 2$, parameters of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ are partially shared via repetitions of $\boldsymbol{g}$.

Before presenting the numerical results, let us first present the hyper-parameters for training our network architecture $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$. For each $i \in \{0, 1\}$, we the uniform distribution to randomly choose 3×10^5 training samples and 3×10^4 test samples in $\mathcal{S}_i$, associated with the same label i. Then, we use these 6×10^5 training samples to train the networks and use these 6×10^4 test samples to compute the test accuracy. We adopt RAdam (Liu et al., 2020) as the optimization method. The batch size, the number of epochs, and the learning rate in epochs $5(i-1)+1$ to $5i$ for $i = 1, 2, \cdots, 100$ are set to 512, 500, and $0.002 \times 0.9^{i-1}$, respectively. We add the softmax activation function to the network output and use the cross-entropy loss function to estimate the model loss between the target function and the network output. In our experiment, we standardize all training (test) samples, i.e., we rescale all training (test) samples to make them have a mean of 0 and a standard deviation of 1. To make the experiment more reliable, we repeat our experiment 12 times and discard 2 top-performing and 2 bottom-performing trials by measuring the performance via the average test accuracy of the last 100 epochs. For each epoch, the average of test accuracies in the rest 8 trials is adopted as the target accuracy.

Finally, we are ready to present the experiment results to compare the numerical performances of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ for $r = 1, 2, 3, 4$. In Table 2, we compare the test accuracies of $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ for $r = 1, 2, 3, 4$. We adopt the average test losses of the last 100 epochs as the target test accuracy.

Table 2. Test accuracy comparison.

$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$	$r = 1$	$r = 2$	$r = 3$	$r = 4$
test accuracy	0.5593	0.7934	0.8716	0.8941

See Figure 5 for a comparison of test accurcies over epochs for $r = 1, 2, 3, 4$. As we can see from Figure 5 and Table 1, the test accuracy can be improved by increasing r, as expected. Such experiment results numerically show that repeated compositions of a single fixed-size ReLU network block can produce super approximation power.

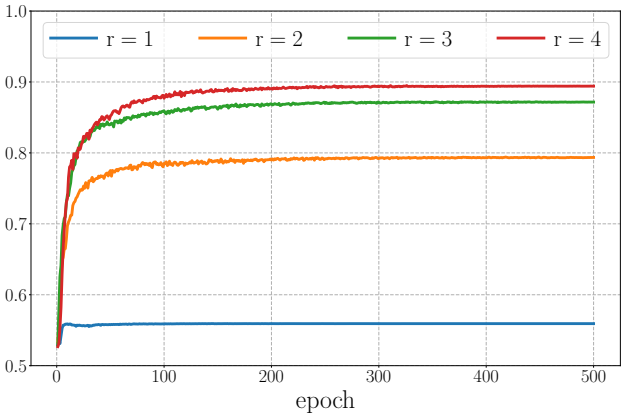


Figure 5. Test accuracies over epochs.

5. Conclusion

This paper studies the expressive power of deep neural networks from the perspective of function compositions. We prove that repeated compositions of a single fixed-size ReLU network can produce super expressive power. As shown in Theorems 1.1 and 1.3, $\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \mathcal{L}_1$ can approximate a continuous functions $f \in C([0, 1]^d)$ with an error $\mathcal{O}\big(\omega_f(r^{-1/d})\big)$, where $\boldsymbol{g}$ is realized by a fixed-size ReLU network, $\mathcal{L}_1$ and $\mathcal{L}_2$ are two affine linear maps matching the dimensions. Furthermore, we discuss the connection between our results and dynamical systems. Our results reveal that a continuous-depth network generated via a dynamical system has good approximation power even if its dynamics function is time-independent and realized by a fixed-size ReLU network.

We remark that our analysis is limited to the ReLU activation function and the fully connected networks. It would be of great interest to extend our results to other activation functions (e.g., the sigmoid and tanh functions) and other network architectures (e.g, convolutional neural networks). Besides, our numerical examples are pretty simple and hence it would be also interesting to further explore the numerical performance of our network architecture and apply it to real-world applications.

Acknowledgements

We would like to thank Zuowei Shen, Yimin Zhong, and Haomin Zhou for their constructive comments on our theoretical proofs and numerical experiments.

References

Bao, C., Li, Q., Shen, Z., Tai, C., Wu, L., and Xiang, X. Approximation analysis of convolutional neural networks. *Semantic Scholar e-Preprint*, art. Corpus ID:

204762668, 2019.

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. doi: 10.1109/18.256500.

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

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. doi: 10.1007/BF02551274.

E, W. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 2017.

Gribonval, R., Kutyniok, G., Nielsen, M., and Voigtlaender, F. Approximation spaces of deep neural networks. *Constructive Approximation*, 55:259–367, 2022.

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. doi: 10.1109/CVPR.2016.90.

Holmes, P. History of dynamical systems. *Scholarpedia*, 2(5):1843, 2007. doi: 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/089360809190009T.

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.

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. doi: 10.48550/arXiv.2208.08707.

Li, Q., Lin, T., and Shen, Z. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, to appear. doi: 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. doi: 10.48550/arXiv.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. doi: 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.

Savarese, P. and Maire, M. Learning implicitly recurrent CNNs through parameter sharing. In *International Conference on Learning Representations*, 2019.

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. doi: 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. doi: 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 *Advances in Neural Information Processing Systems*, 2022.

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. doi: 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.

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. doi: 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.

A. Proofs of Theorem 1.1 and 1.3

In this section, we will prove Theorem 1.1 and 1.3 based on an auxiliary theorem, Theorem A.1, which will be proved in Section B. Notations throughout this paper are summarized in Section A.1.

A.1. Notations

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

- Let $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{Z}$ denote the set of real numbers, rational numbers, and integers, respectively.
- Let $\mathbb{N}$ and $\mathbb{N}^+$ denote the set of natural numbers and positive natural numbers, respectively. That is, $\mathbb{N}^+ = \{1, 2, 3, \cdots\}$ and $\mathbb{N} = \mathbb{N}^+ \cup \{0\}$.
- For any $x \in \mathbb{R}$, let $\lfloor x \rfloor := \max\{n : n \le x,\ n \in \mathbb{Z}\}$ and $\lceil x \rceil := \min\{n : n \ge x,\ n \in \mathbb{Z}\}$.
- Let $\mathbb{1}_S$ be the indicator (characteristic) function of a set S, i.e., $\mathbb{1}_S$ is equal to 1 on S and 0 outside S.
- The set difference of two sets A and B is denoted by $A \backslash B := \{x : x \in A,\ x \notin B\}$.
- Matrices are denoted by bold uppercase letters. For instance, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a real matrix of size $m \times n$ and $\boldsymbol{A}^T$ denotes the transpose of $\boldsymbol{A}$. Vectors are denoted as bold lowercase letters. For example, $\boldsymbol{a} = (a_1, \cdots, a_d) \in \mathbb{R}^d$.
- For any vector $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$, we use $[\boldsymbol{x}]_{[n:m]}$ to denote a slice of $\boldsymbol{x}$ for its n-th to m-th entries and denote $[\boldsymbol{x}]_{[n]}$ as 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 $[6\boldsymbol{x} + 1]_{[3]} = 6x_3 + 1$.
- For any $p \in [1, \infty)$, the p-norm (or ℓ^p-norm) of a vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$ is defined by
$$\|\boldsymbol{x}\|_p = \|\boldsymbol{x}\|_{\ell^p} := \big(|x_1|^p + |x_2|^p + \cdots + |x_d|^p\big)^{1/p}.$$
In the case of $p = \infty$,
$$\|\boldsymbol{x}\|_\infty = \|\boldsymbol{x}\|_{\ell^\infty} := \max\big\{|x_i| : i = 1, 2, \cdots, d\big\}.$$
- By convention, $\sum_{j=n_1}^{n_2} a_j = 0$ if $n_1 > n_2$, 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 $\delta \in (0, \frac{1}{K})$, define a trifling region $\Omega([0, 1]^d, K, \delta)$ of $[0, 1]^d$ as
$$\Omega([0, 1]^d, K, \delta) := \bigcup_{j=1}^{d} \left\{ \boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in [0, 1]^d : x_j \in \bigcup_{k=1}^{K-1} \left(\tfrac{k}{K} - \delta, \tfrac{k}{K}\right) \right\}. \tag{5}$$
In particular, $\Omega([0, 1]^d, K, \delta) = \emptyset$ if $K = 1$. See Figure 6 for two examples of trifling regions.
- Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote the rectified linear unit (ReLU), i.e. $\sigma(x) = \max\{0, x\}$ for any $x \in \mathbb{R}$. With a slight abuse of notation, we define $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ as $\sigma(\boldsymbol{x}) = \big(\sigma(x_1), \cdots, \sigma(x_d)\big)$ for any $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$.
- A (vector-valued) function $\boldsymbol{\phi}$ realized by a ReLU network can be briefly described as follows:
$$\boldsymbol{x} = \widetilde{\boldsymbol{h}}_0 \xrightarrow[\boldsymbol{\mathcal{L}}_0]{\boldsymbol{W}_0,\, \boldsymbol{b}_0} \boldsymbol{h}_1 \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_1 \quad \cdots \quad \xrightarrow[\boldsymbol{\mathcal{L}}_{L-1}]{\boldsymbol{W}_{L-1},\, \boldsymbol{b}_{L-1}} \boldsymbol{h}_L \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_L \xrightarrow[\boldsymbol{\mathcal{L}}_L]{\boldsymbol{W}_L,\, \boldsymbol{b}_L} \boldsymbol{h}_{L+1} = \boldsymbol{\phi}(\boldsymbol{x}),$$
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 transformation $\boldsymbol{\mathcal{L}}_i$, 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
$$\widetilde{\boldsymbol{h}}_i = \sigma(\boldsymbol{h}_i) \quad \text{for } i = 1, 2, \cdots, L.$$

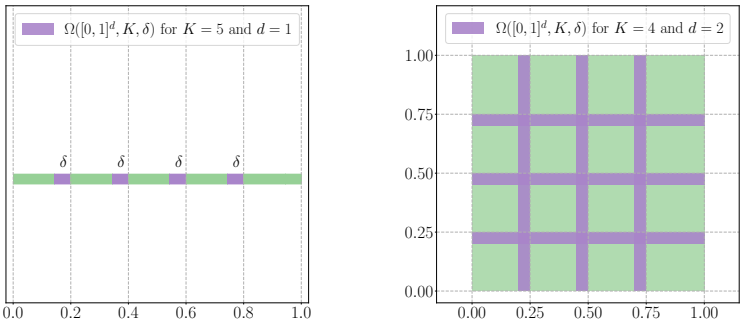


Figure 6. Two examples of trifling regions. (a) $K = 5, d = 1$. (b) $K = 4, d = 2$.

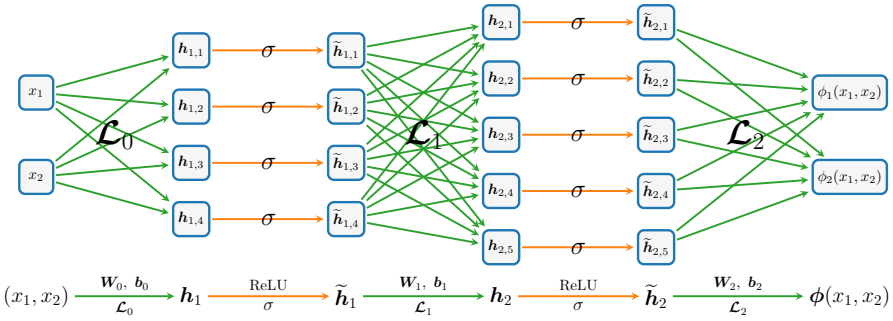


Figure 7. An example of a ReLU network of width 5 and depth 2. A vector-valued function $\boldsymbol{\phi} = (\phi_1, \phi_2)$ is realized by this network.

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

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

which has been illustrated in Figure 7.

- The expression "a network of width N and depth L" means
  - The number of neurons in each **hidden** layer of this network (architecture) is no more than N.
  - The number of **hidden** layers of this network (architecture) is no more than L.

A.2. Proof of Theorems 1.1

In the proof of Theorems 1.1 (or 1.3), we essentially construct a piecewise constant function to approximate the target continuous function. However, the continuity of ReLU networks makes them unable to uniformly approximate piecewise constant functions well. Thus, we introduce the trifling region $\Omega([0,1]^d, K, \delta)$, defined in Equation (5), and use ReLU networks to realize piecewise constant functions outside the trifling region. To simplify the proof of Theorems 1.1 (or 1.3), we introduce an auxiliary theorem, Theorem A.1 below. As we shall see later, Theorem A.1 is a key intermediate step in the proof of Theorems 1.1 (or 1.3).

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

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

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

The proof of Theorem A.1 is placed in Section B. Now, let we prove Theorem 1.1 by assuming Theorem A.1 is true, which will be proved in Section B.

Proof of Theorem 1.1. 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 let $\delta \in (0, \frac{1}{3K}]$ be an arbitrary number determined later. By Theorem A.1, there exist $\boldsymbol{g}_1 \in \mathcal{NN}\{39d+24,\, 3;\ \mathbb{R}^{5d+3} \to \mathbb{R}^{5d+3}\}$ and $\widehat{\boldsymbol{\mathcal{L}}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ and $\widehat{\mathcal{L}}_2 : \mathbb{R}^{5d+3} \to \mathbb{R}$ such that

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

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 $\widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1(\boldsymbol{x}) - f$, we need to further bound it inside $\Omega([0,1]^d, K, \delta)$. To this end, we define

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

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 $\boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, r, \delta)$, 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.

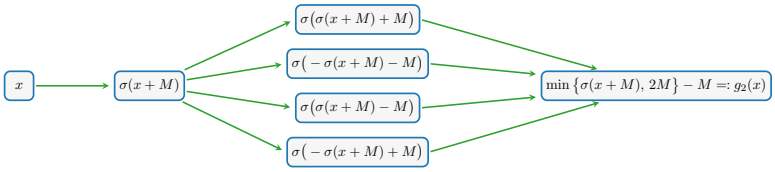


Figure 8. An illustration of the ReLU network realizing g_2.

Let $\widehat{\mathcal{L}}_3 : \mathbb{R} \to \mathbb{R}$ as the identity map. Then, by Proposition 3.3 with $N_1 = 9d+24$, $N_2 = 4$, $L_1 = 3$, $L_2 = 2$, $d_0 = d$, $d_1 = 5d+3$, $d_2 = d_3 = 1$ therein and setting $\widetilde{d} = 5d+3$, 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 $\boldsymbol{\mathcal{L}}_1 : \mathbb{R}^d \to \mathbb{R}^{5d+5}$ 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 $\left\|g_2 \circ \widehat{\mathcal{L}}_2 \circ \boldsymbol{g}_1^{\circ(3r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1\right\|_{L^\infty(\mathbb{R}^d)} \leq M$ and

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

Thus, we have

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

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

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

for any $\boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, r, \delta)$.

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

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

we have

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

Hence, we have $\|\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(3r+1)} \circ \boldsymbol{\mathcal{L}}_1 - f\|_{L^p([0,1]^d)} = \|\phi - f\|_{L^p([0,1]^d)} \leq 6\sqrt{d}\,\omega_f(r^{-1/d})$. So we finish the proof of Theorem 1.1 □

A.3. Proof of Theorems 1.3

We will apply Theorem A.1 to prove Theorem 1.3. Note that the approximation error inside the trifling region is not bounded. To prove Theorem 1.3 with an pointwise approximation, we must control the approximation error inside the trifling region. To this end, we introduce a theorem to handle the approximation inside 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*

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

Then

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

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 ready to give the detailed proof of Theorem 1.3.

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 $\boldsymbol{\mathcal{L}}_{0,1} : \mathbb{R}^d \to \mathbb{R}^{5d+3}$ 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

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

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.

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, We claim that 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

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

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

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

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

$$\widehat{\boldsymbol{G}}(x_1,x_2,x_3)\coloneqq\Big(\mathrm{mid}(x_1,x_2,x_3),\ 0,\ 0\Big)\quad\text{for any }(x_1,x_2,x_3)\in\mathbb{R}^3,$$

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

$$\widehat{\mathcal{L}}_3(x_1,x_2,x_3)\coloneqq x_1\quad\text{for any }(x_1,x_2,x_3)\in\mathbb{R}^3.$$

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

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

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), $\mathrm{mid}(\cdot,\cdot,\cdot)$ 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 $\widetilde{d}=\max\{3d_j,3\}=3d_j$, there exist

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

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

$$\Big|\mathcal{L}_2\circ\boldsymbol{g}^{\circ(3r+2d-1)}\circ\boldsymbol{\mathcal{L}}_1(\boldsymbol{x})-f(\boldsymbol{x})\Big|=|\phi(\boldsymbol{x})-f(\boldsymbol{x})|\le6\sqrt{d}\,\omega_f(r^{-1/d}).$$

Thus, we finish the proof of Theorem 1.3. □

B. Proof of auxiliary theorem

In this section, we will prove the auxiliary theorem, Theorem A.1, based on Propositions 3.1, 3.2, and 3.3, the proof of which can be found in Sections C, D, and E, respectively. Now, let us give the proof of Theorem A.1 by assuming these three propositions are true.

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 $\Omega([0,1]^d, K, \delta)$ and 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 (5).
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 $\phi_2 = \widehat{\mathcal{L}}_5 \circ \boldsymbol{g}_2^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_4 \circ \widehat{\boldsymbol{\mathcal{L}}}_3$ 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 $\widehat{\boldsymbol{\mathcal{L}}}_3(\boldsymbol{\beta})$ approximate to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Then, $\phi_2 = \widehat{\mathcal{L}}_5 \circ \boldsymbol{g}_2^{\circ(r-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_4 \circ \widehat{\boldsymbol{\mathcal{L}}}_3$ maps $\boldsymbol{\beta}$ approximate 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 $\boldsymbol{x}_{\boldsymbol{\beta}} \coloneqq \boldsymbol{\beta}/K$ and

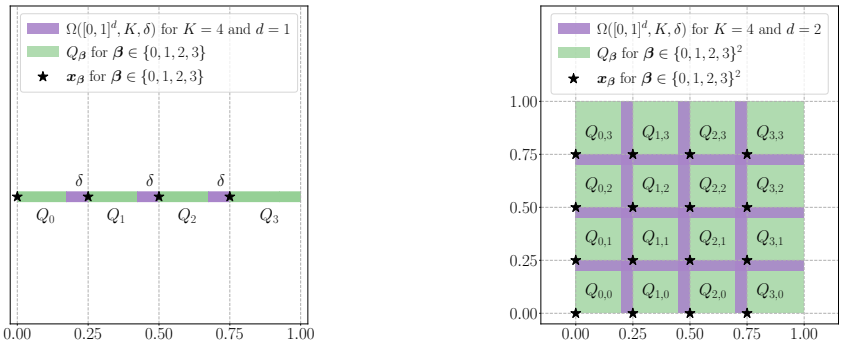
$$Q_{\boldsymbol{\beta}} \coloneqq \left\{\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in [0,1]^d : x_i \in \left[\tfrac{\beta_i}{K}, \tfrac{\beta_i+1}{K} - \delta \cdot \mathbb{1}_{\{\beta_i \le K-2\}}\right], \quad i = 1,2,\cdots,d\right\}.$$

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

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

where $\Omega([0,1]^d, K, \delta)$ is the trifling region defined in Equation (5). See Figure 9 for illustrations of $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$ and $\Omega([0,1]^d, K, \delta)$.

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



(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 $d = 2$.

Set $\widetilde{\delta} = K\delta$. By Proposition 3.1 with $m = r$ and $n = K = \lfloor r^{1/d} \rfloor \le r = m$, 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

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

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

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

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

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

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

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

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

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

to construct a continuous piecewise linear function $h : [0, 1] \to \left[0,\, 2\omega_f(\sqrt{d})\right]$, where $\mathbf{1} = (1, \cdots, 1) \in \mathbb{R}^d$. 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 $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

- 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. It is easy to verify that

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

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

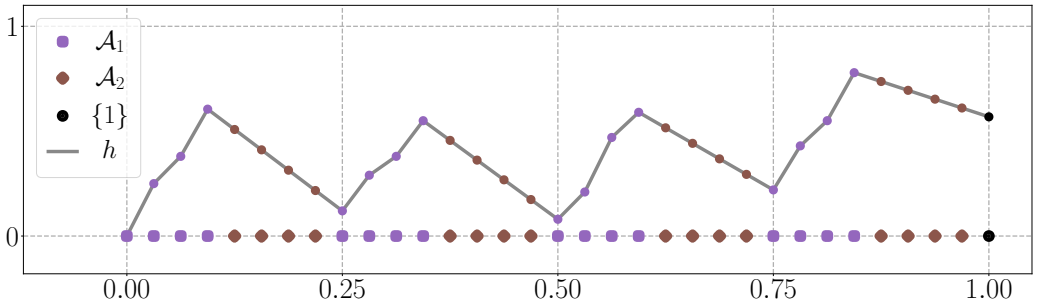


Figure 10. An illustration of $\mathcal{A}_1$, $\mathcal{A}_2$, $\{1\}$, and h for $K = 4$ and $d = 2$.

By Proposition 3.2 with $y_j = h(\frac{j}{2K^d})$, $\varepsilon = \omega_f(\frac{\sqrt{d}}{K}) > 0$, $m = 2r$, and $n = 2K^d = 2\lfloor K^{1/d}\rfloor^d \le 2r = m$ 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

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

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

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

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

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

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

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

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

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

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

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 $\widetilde{d} = 5d + 1 \ge \max\{5d,\ 6\}$, there exist

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

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

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

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

Recall that $K = \lfloor r^{1/d} \rfloor \ge \frac{r^{1/d}}{2}$, $\omega_f(t) = \omega_{\widetilde{f}}(t)$, and $\omega_f(n \cdot t) \le n \cdot \omega_f(t)$ for any $n \in \mathbb{N}^+$ and $t \in [0,\infty)$. 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

$$\begin{aligned}
|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| &\le \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K}) \le 2\omega_f(\tfrac{\sqrt{d}}{K}) = 2\omega_f(\tfrac{\sqrt{d}}{\lfloor r^{1/d} \rfloor}) \\
&\le 2\omega_f(2\sqrt{d}\, r^{-1/d}) \le 2\big\lceil 2\sqrt{d}\,\big\rceil \omega_f(r^{-1/d}) \le 5\sqrt{d}\,\omega_f(r^{-1/d}),
\end{aligned}$$

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

C. Proof of Proposition 3.1

The key point of proving Proposition 3.1 is the composition architecture of neural networks. To simplify the proof, we first establish a lemma, Lemma C.1 below, which indeed can be regarded as a weak variant of Proposition 3.1.

Lemma C.1. *Given any $\delta \in (0,1)$ and $n \in \mathbb{N}^+$ with $n \ge 2$, 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*

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

The proof of Lemma C.1 is place later in this section. Now, we are ready to give the detailed proof of Proposition 3.1 by assuming Lemma C.1 is true.

Proof of Proposition 3.1. We may assume $n \ge 2$ 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

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

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

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

Therefore, by Equations (11) and (12), we have

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \boldsymbol{\mathcal{L}}_1(x) = \mathcal{L}_2 \circ \boldsymbol{g}^{\circ(m-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1\Big(\mathcal{L}_0(x)\Big) = \big\lfloor \mathcal{L}_0(x) \big\rfloor = k$$

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

It remains to prove Equation (12). To this end, we only need to show

$$k \le \mathcal{L}_0(k) \le \mathcal{L}_0\big(k+1-\delta\cdot \mathbb{1}_{\{k\le n-2\}}\big) \le k+1-\widetilde{\delta} \quad \text{for } k=0,1,\cdots,n-1. \tag{13}$$

Clearly, $\mathcal{L}_0$ is increasing. To prove Equation (13), we only need to prove the following two equations:

$$k \le \mathcal{L}_0(k) \quad \text{for } k=0,1,\cdots,n-1 \tag{14}$$

and

$$\mathcal{L}_0\big(k+1-\delta\cdot \mathbb{1}_{\{k\le n-2\}}\big) \le k+1-\widetilde{\delta} \quad \text{for } k=0,1,\cdots,n-1. \tag{15}$$

Let us first prove Equation 14. Clearly, for $k=0,1,\cdots,n-1$, we have

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

where the inequality comes from the fact $n(n-k)-(1-\delta)k \ge n-k \ge 0$.

Next, let us prove Equation 15. 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

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

So we finish the proof of Proposition 3.1. □

Let us prove Lemma C.1 to end this section.

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 $x\in\mathbb{R}$. 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 $x-k+\delta\ge 0$, $x-k\ge 0$, $-x+k+1\ge 0$, and $-x+k+1-\delta\ge 0$, 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 $x-k+\delta\le 0$, $x-k\le 0$, $-x+k+1\ge 0$, and $-x+k+1-\delta\ge 0$, 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

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

See an illustration of h_k in Figure 11.

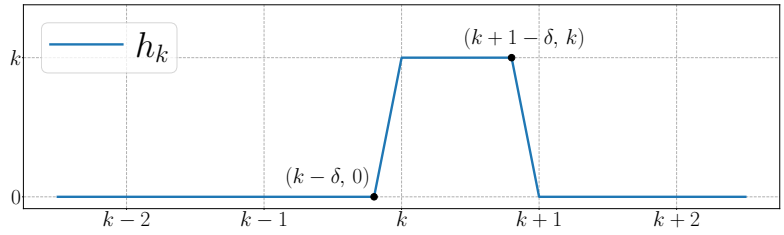


Figure 11. An illustration of h_k.

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

$$h\big(kx, k^2, k\big) = \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta} + k\big) - \sigma\big(\tfrac{kx}{\delta} - \tfrac{k^2}{\delta}\big) + \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta}\big) - \sigma\big(-\tfrac{kx}{\delta} + \tfrac{k^2}{\delta} + \tfrac{k}{\delta} - k\big) - k = h_k(x) \quad (16)$$

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

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

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

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, $\boldsymbol{g} \in \mathcal{NN}\{9,\ 1;\ \mathbb{R}^5 \to \mathbb{R}^5\}$.

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

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

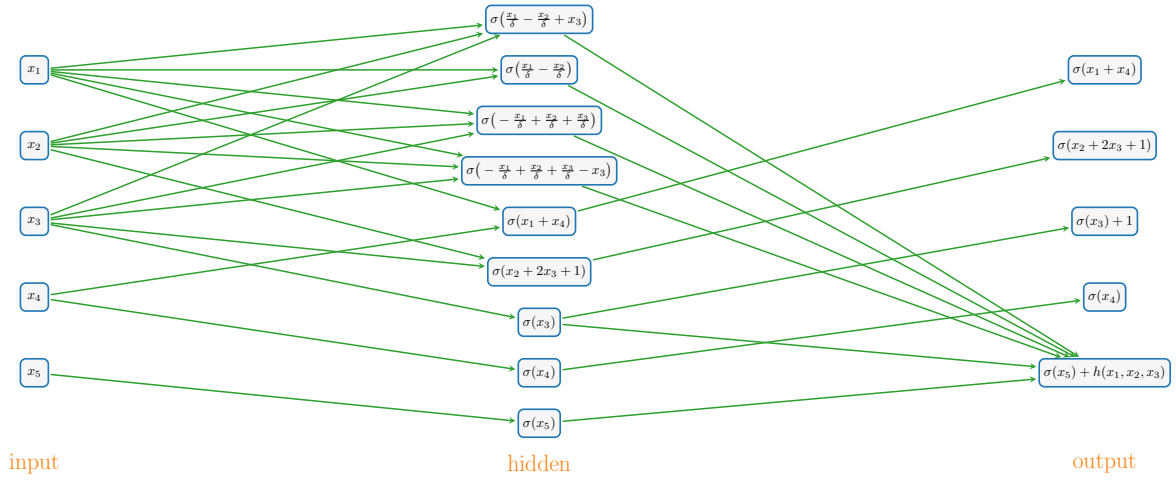


Figure 12. An illustration of $\boldsymbol{g} : \mathbb{R}^5 \to \mathbb{R}^5$.

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

$$\begin{aligned}
\boldsymbol{g}(\boldsymbol{\xi}_k) &= \boldsymbol{g}\left(kx,\ k^2,\ k,\ x,\ \sum_{i=0}^{k-1} h_i(x)\right) \\
&= \left(\sigma(kx + x), \quad \sigma(k^2 + 2k + 1), \quad \sigma(k) + 1, \quad \sigma(x), \quad \sigma\left(\sum_{i=0}^{k-1} h_i(x)\right) + \underbrace{h(kx, k^2, k)}_{=h_k(x) \text{ by (16)}}\right) \\
&= \left((k+1)x, \quad (k+1)^2, \quad (k+1), \quad x, \quad \sum_{i=0}^{k} h_i(x)\right) = \boldsymbol{\xi}_{k+1},
\end{aligned}$$

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

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 key idea of proving Proposition 3.2 is the bit extraction technique proposed in (Bartlett et al., 1998). Before proving Proposition 3.2, we establish a lemma, Lemma E.1, which is 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 $\theta_1, \theta_2, \cdots, \theta_r \in \{0, 1\}$, we have*

$$\mathcal{L}_2 \circ \boldsymbol{g}^{\circ r} \circ \boldsymbol{\mathcal{L}}_1\big(k,\ \text{bin}\, 0.\theta_1\theta_2\cdots\theta_r\big) = \sum_{\ell=1}^{k} \theta_\ell \quad \text{for } k = 0, 1, \cdots, r. \tag{17}$$

The proof of Lemma D.1 is placed later in this section. Now we are ready to give the proof of Proposition 3.2 by assuming Lemma D.1.

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

$$\begin{aligned} a_k &= a_0 + \sum_{i=1}^{k} c_i - \sum_{i=1}^{k} d_i \\ &= a_0 + \widetilde{\mathcal{L}}_2 \circ \widetilde{\boldsymbol{g}}^{\circ(n-1)} \circ \widetilde{\boldsymbol{\mathcal{L}}}_1\big(k,\, \text{bin}\, 0.c_1 \cdots c_{n-1}\big) - \widehat{\mathcal{L}}_2 \circ \widehat{\boldsymbol{g}}^{\circ(n-1)} \circ \widehat{\boldsymbol{\mathcal{L}}}_1\big(k,\, \text{bin}\, 0.d_1 \cdots d_{n-1}\big). \end{aligned} \tag{18}$$

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

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

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

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

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

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

where the last equality come from Equation (18). 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. □

Let us give the proof of Lemma D.1 to end this section.

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

$$\mathcal{T}(x) = \sigma\big(\tfrac{x}{\delta} + 1\big) - \sigma\big(\tfrac{x}{\delta}\big) \quad \text{for any } x \in \mathbb{R}. \tag{19}$$

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

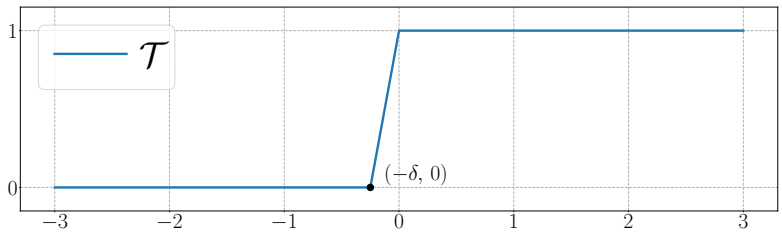


Figure 13. An illustration of $\mathcal{T}$.

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

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

It is easy to verify that

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

implying

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

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 $xy = \sigma(x + y - 1)$ for any $x, y \in \{0, 1\}$ implies

$$\begin{aligned}\sum_{i=1}^{k} \theta_i = \sum_{i=1}^{k} \theta_i + \sum_{i=k+1}^{r} 0 = \sum_{i=1}^{r} \theta_i \cdot \mathcal{T}(k-i) &= \sum_{i=1}^{r} \sigma\Big(\theta_i + \mathcal{T}(k-i) - 1\Big) \\ &= \sum_{i=1}^{r} \sigma\Big(\mathcal{T}(\beta_i - \tfrac{1}{2}) + \mathcal{T}(k-i) - 1\Big).\end{aligned} \tag{20}$$

Define $\boldsymbol{g} : \mathbb{R}^3 \to \mathbb{R}^3$ 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 $\boldsymbol{\xi}_{r+1} = \boldsymbol{g}^{\circ r}(\boldsymbol{\xi}_1)$.

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

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

from which we deduce

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

where the last equality comes from Equation (20).

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

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

We will prove Proposition 3.3 in this section. To simplify the proof, let us introduce two lemmas below.

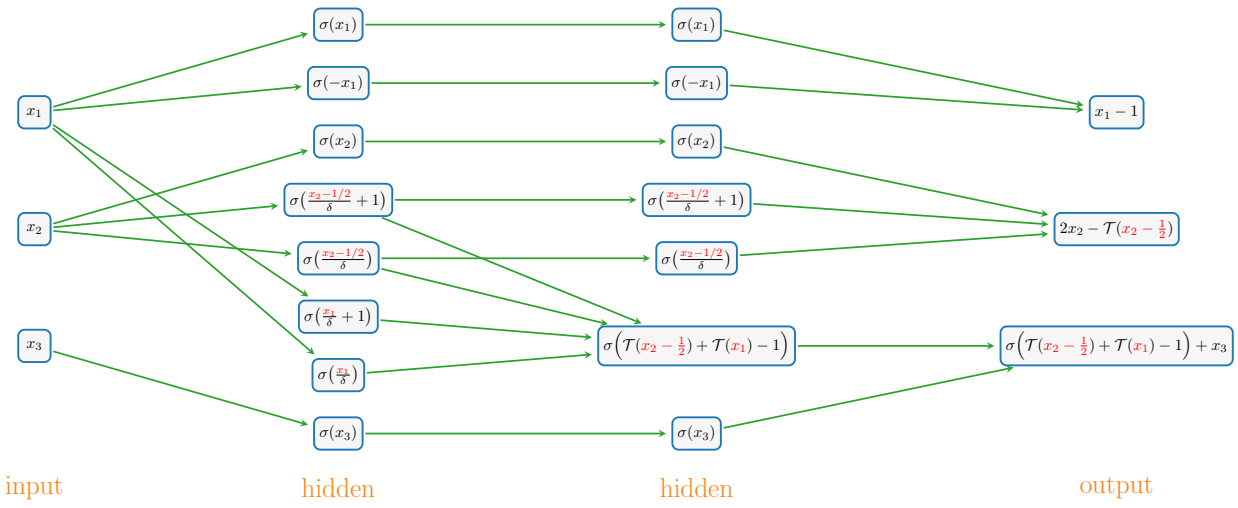


Figure 14. An illustration of the ReLU network realizing $\boldsymbol{g}(x_1, x_2, x_3)$ for $(x_1, x_2, x_3) \in \mathbb{R} \times [0, \infty)^2$ based on Equation (19) and the fact $t = \sigma(t) - \sigma(-t)$ for any $t \in \mathbb{R}$.

Lemma E.1. *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 $t \in (-\infty, -1] \cup [1, \infty)$.

Lemma E.2. *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 $\boldsymbol{x} \in [-A, A]^d$.

The proofs of Lemma E.1 and E.2 are placed later in this section. Now, let us present the proof of Proposition 3.3 by assuming Lemma E.1 and E.2 are true.

Proof of Proposition 3.3. Set

$$\widetilde{A} = 100(r_1 + r_2 + 1) + \sup_{\boldsymbol{x} \in [-A, A]^{d_0}} \big\|\widetilde{\boldsymbol{\mathcal{L}}}_0(\boldsymbol{x})\big\|_{\ell^\infty}.$$

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.2, 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

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

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.2, 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

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

So we finish the proof of Proposition 3.3. □

Now let us give the proof of Lemma E.1.

Proof of Lemma E.1. The key idea of 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 $u, v \in \mathbb{R}$ and $t \in (-\infty, -1] \cup [1, \infty)$. To this end, we define

$$g(u, v, t) \coloneqq \sigma\big(u + Mt\big) + \sigma\big(v - Mt\big) - M\sigma(t) - M\sigma(-t). \tag{21}$$

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

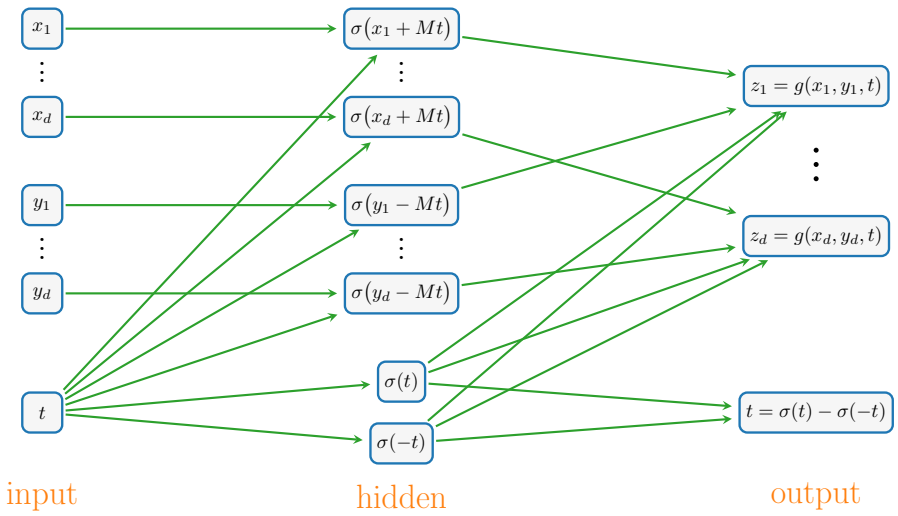


Figure 15. An illustration of the target ReLU network realizing $\boldsymbol{\phi}$ based on Equation (21).

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.1. □

Let us prove Lemma E.2 to end this section.

Proof of Lemma E.2. 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.1 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 $\boldsymbol{a}=(a_1,a_2,\cdots,a_d)\in\mathbb{R}^d$:

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

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

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

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.1, there exists

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

such that

$$\boldsymbol{\psi}(\boldsymbol{u},\boldsymbol{v},t)=(\boldsymbol{w},t)\quad\text{with}\quad\boldsymbol{w}=\begin{cases}\boldsymbol{u} & \text{if } t\geq 1\\ \boldsymbol{v} & \text{if } t\leq -1\end{cases}\tag{22}$$

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, $\boldsymbol{G} \in \mathcal{NN}\{\max\{N_1, d\} + \max\{N_2, d\} + 1,\ \max\{L_1, L_2\};\ \mathbb{R}^{d+1} \to \mathbb{R}^{2d+1}\}$ and $\boldsymbol{\psi} \in \mathcal{NN}\{2d+2,\ 1;\ \mathbb{R}^{2d+1} \to \mathbb{R}^{d+1}\}$ implies

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

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

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

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

$$\big(\boldsymbol{\xi}_k,\ t_k\big) = \boldsymbol{\Phi}(\boldsymbol{\xi}_{k-1}, t_{k-1}) = \boldsymbol{\psi} \circ \boldsymbol{G}(\boldsymbol{\xi}_{k-1}, t_{k-1}) = \boldsymbol{\psi}\Big(\boldsymbol{g}_1(\boldsymbol{\xi}_{k-1}),\ \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1}),\ g_0(t_{k-1})\Big).$$

Then, by Equation (22), we have

$$t_k = g_0(t_{k-1}) = t_{k-1} - 2 \quad \text{and} \quad \boldsymbol{\xi}_k = \begin{cases} \boldsymbol{g}_1(\boldsymbol{\xi}_{k-1}) & \text{if } t_k = g_0(t_{k-1}) \geq 1 \\ \boldsymbol{g}_2(\boldsymbol{\xi}_{k-1}) & \text{if } t_k = g_0(t_{k-1}) \leq -1 \end{cases}$$

for $k = 1, 2, \cdots, r_1 + r_2$, implying $t_k = 2r_1 + 1 - 2k = 2(r_1 - k) + 1$.

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

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

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

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

That means

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

Moreover,

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

Thus, we have

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

which means we complete the proof of Lemma E.2. □