Journal of Machine Learning Research 26 (2025) 1-76 Submitted 8/24; Revised 1/25; Published 4/25

DRM Revisited: A Complete Error Analysis

Yuling Jiao YULINGJIAOMATH@WHU.EDU.CN
School of Artificial Intelligence, Wuhan University, Wuhan, China
National Center for Applied Mathematics in Hubei, Wuhan University, Wuhan, China
Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan, China

Ruoxuan Li RUOXUANLI.MATH@WHU.EDU.CN
School of Mathematics and Statistics, Wuhan University, Wuhan, China

Peiying Wu PEIYINGWU@WHU.EDU.CN
School of Mathematics and Statistics, Wuhan University, Wuhan, China

Jerry Zhijian Yang* ZJYANG.MATH@WHU.EDU.CN
School of Mathematics and Statistics, Wuhan University, Wuhan, China
Wuhan Institute for Math & AI, Wuhan University, Wuhan, China
National Center for Applied Mathematics in Hubei, Wuhan University, Wuhan, China
Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan, China

Pingwen Zhang PWZHANG@WHU.EDU.CN
Wuhan Institute for Math & AI, Wuhan University, Wuhan, China
School of Mathematical Sciences, Peking University, Beijing, China

Editor: Jianfeng Lu

Abstract

It is widely known that the error analysis for deep learning involves approximation, statistical, and optimization errors. However, it is challenging to combine them together due to overparameterization. In this paper, we address this gap by providing a comprehensive error analysis of the Deep Ritz Method (DRM). Specifically, we investigate a foundational question in the theoretical analysis of DRM under the overparameterized regime: given a target precision level, how can one determine the appropriate number of training samples, the key architectural parameters of the neural networks, the step size for the projected gradient descent optimization procedure, and the requisite number of iterations, such that the output of the gradient descent process closely approximates the true solution of the underlying partial differential equation to the specified precision?

Keywords: deep Ritz method, projected gradient descent, over-parameterization, complete error analysis, new optimization error analysis

1. Introduction

Classical numerical methods, such as finite element methods (Brenner and Scott, 2007; Ciarlet, 2002), face difficulties when solving high-dimensional partial differential equations (PDEs). The success of deep learning methods in high-dimensional data analysis has led to the development of promising approaches for solving high-dimensional PDEs using deep neural networks, which have attracted much attention (Anitescu et al., 2019; Sirignano

*. Corresponding author.

and Spiliopoulos, 2018; Lu et al., 2021b; Raissi et al., 2019; Weinan and Yu, 2017; Zang et al., 2020; Berner et al., 2020; Han et al., 2018; Liao and Ming, 2021). Due to the excellent approximation power of deep neural networks, several numerical schemes have been proposed for solving PDEs, including physics-informed neural networks (PINNs) (Raissi et al., 2019), weak adversarial networks (WAN) (Zang et al., 2020) and the deep Ritz method (Weinan and Yu, 2017). PINNs is based on residual minimization, while WAN is inspired by Galerkin method. Based on classical Ritz method, the deep Ritz method is proposed to solve variational problems corresponding to a class of PDEs, which has become one of the most renowned approaches in the field of elliptic equations.

The success of deep learning methods in solving high dimensional PDEs has propelled the advancement of its theoretical research. It is now widely recognized that, as a non-parametric estimation method, error analysis in deep learning for PDEs includes approximation error, statistical error (also called generalization error), and optimization error (Grohs and Kutyniok, 2022; Telgarsky, 2021; Weinan, 2020; Bach, 2023). To date, the existing convergence analysis for these deep solvers has predominantly focused on characterizing the trade-offs between the approximation error and the statistical error (Weinan et al., 2019; Hong et al., 2021; Lu et al., 2021d; Hutzenthaler et al., 2020; Shin, 2020; Lanthaler et al., 2022; Müller and Zeinhofer, 2021; Mishra and Rusch, 2021; Kutyniok et al., 2022; Son et al., 2021; Wang et al., 2022; Weinan et al., 2020; Jiao et al., 2022; Duan et al., 2022; Lu et al., 2021c; Mishra and Molinaro, 2022; Ji et al., 2024; Yang and He, 2024; Hu et al., 2023, 2024; Dai et al., 2023; Yu et al., 2024). Meanwhile, these results are conducted in scenarios where the number of neural network parameters is smaller than the number of training samples. However, in practical applications, over-parameterized networks, where the number of parameters far exceeds the number of samples, are more commonly used since empirical evidence suggests that over-parameterization makes the training computationally more efficient. Moreover, recent theoretical studies have indicated that the training loss will converges to zero linearly if one properly initialized the (stochastic) gradient descent specialized in over-parameterized regimes, even though the optimization problem is highly non-convex (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou and Gu, 2019; Liu et al., 2022; Chizat et al., 2019).

The fundamental drivers behind the empirical success of over-parameterized deep learning models continue to elude full understanding, especially when simultaneously accounting for the complex interplay between approximation, generalization, and optimization (Belkin, 2021; Bartlett et al., 2021; Berner et al., 2022). Extensive research efforts have been dedicated to elucidating the role of over-parameterization in linear and kernel models, particularly from the perspective of the double descent phenomenon (Belkin et al., 2018, 2019a; Hastie et al., 2022; Belkin et al., 2019b; Liang and Rakhlin, 2020; Nakkiran et al., 2020; Bartlett et al., 2020; Tsigler and Bartlett, 2023; Belkin, 2021; Bartlett et al., 2021; Tsigler and Bartlett, 2023). However, a crucial gap remains in providing a comprehensive error analysis that jointly accounts for all three key error components. This challenge persists even for the empirical risk minimization estimator in over-parameterized deep learning settings, which has been shown to potentially yield inconsistent results (Kohler and Krzyzak, 2021).

1.1 Contributions

- In this work, we establish the first comprehensive error analysis for the deep Ritz method in the over-parameterized setting. This analysis jointly accounts for all three key error components: approximation error, statistical error, and optimization error.

- Technically, we derive a novel error decomposition, where the optimization error term we employ is distinct from and tighter than those used in the prior literature. This error decomposition is of independent theoretical interest and holds value for the analysis of other deep learning tasks.

- Unlike previous analyses of optimization error, a key feature of our main results is that they do not require the entire training dynamics to remain confined within an infinitesimally small neighborhood of the initial parameter values. This reduces the gap between theory and practical training.

1.2 Organizations

The paper is organized as follows. In Section 2, we first introduce the notation, the parallel neural network architecture $\mathcal{PNN}$, and the projected gradient descent (PGD) algorithm used for optimization. Then, following these preliminaries, we present the main theorem of this paper. In Section 3, we present the proof of the main theorem, divided into five subsections covering a novel error decomposition method, approximation error bounds, statistical error estimates, optimization error control, and the combination of all the separate analysis. In Section 4, we discuss related work in detail and highlight our contributions. Finally, in Section 5, we provide a summary of the paper and outline our planned future work. All proof details are provided in the appendix.

2. Main Result

In this section, we present the main theoretical result of this paper, which establishes the first comprehensive error analysis for the deep Ritz method in the over-parameterized setting. To achieve this, we first need some groundwork: In Sections 2.1 and 2.2, we introduce necessary notation and the parallel neural network class $\mathcal{PNN}$ used in this paper. In Section 2.3, we review the deep Ritz method. In Section 2.4, we provide a detailed exposition of the employed optimization algorithm: the projected gradient descent algorithm. Finally, in Section 2.5, we formally propose our main result.

2.1 Notation

In this section, we provide all the notation needed in this paper. We use bold-faced letters to denote vectors and capital letters to denote matrices or fixed parameters. Unless otherwise specified, C represents a constant, and $C(a, b)$ or $C_i(a, b)$ represents functions that depend only on a and b. If $f : X \to Y$ and $g : Y \to Z$ are two functions, then $g \circ f : X \to Z$ represents their composition. For two positive functions $f(x)$ and $g(x)$, the asymptotic notation $f(x) = \mathcal{O}(g(x))$ denotes $f(x) \leq Cg(x)$ for some constant $C > 0$. The notation $\widetilde{\mathcal{O}}(\cdot)$ is used to ignore logarithmic terms.

Let $\mathbb{N}$ denote the set of natural numbers. We define $\mathbb{N}^+ := \{x \in \mathbb{N} \,|\, x > 0\}$. If $x \in \mathbb{R}$, $\lfloor x \rfloor := \max\{k \in \mathbb{N} : k \leq x\}$ denotes the largest integer strictly smaller than x and $\lceil x \rceil := \min\{k \in \mathbb{N} : k \geq x\}$ denotes the smallest integer strictly larger than x. If $N \in \mathbb{N}^+$, we define $[N] := \{1, 2, \ldots, N\}$ to be set of all positive integers less than or equal to N. For a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^{\mathrm{T}} \in \mathbb{R}^d$, the ℓ^2-norm and ℓ^∞-norm of $\boldsymbol{x}$ are, respectively, denoted by $\|\boldsymbol{x}\|_2 := \sqrt{\sum_{i=1}^d x_i^2}$ and $\|\boldsymbol{x}\|_\infty := \max_{1\leq i\leq d} |x_i|$. We use the usual multi-index notation, that is, for $\boldsymbol{\alpha} \in \mathbb{N}^d$ we write $\|\boldsymbol{\alpha}\|_1 := \alpha_1 + \ldots + \alpha_d$ and $\boldsymbol{\alpha}! := \alpha_1! \cdot \ldots \cdot \alpha_d!$.

Let $\Omega \subset \mathbb{R}^d$ be an open set. For a function $f : \Omega \to \mathbb{R}$, the $L^p(\Omega)$ norm of f is defined as

$$\|f\|_{L^p(\Omega)} := \left(\int |f(\boldsymbol{x})|^p \, \mathrm{d}\boldsymbol{x} \right)^{1/p}, \quad p \in (0, \infty); \qquad \|f\|_{L^\infty(\Omega)} := \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})| .$$

We denote the (weak or classical) derivative of order $\boldsymbol{\alpha}$ of f by

$$D^{\boldsymbol{\alpha}} f := \frac{\partial^{\|\boldsymbol{\alpha}\|_1} f}{\partial {x_1}^{\alpha_1} \partial {x_2}^{\alpha_2} \cdots \partial {x_d}^{\alpha_d}} .$$

For $s \in \mathbb{N} \cup \{\infty\}$, we denote by $C^s(\Omega)$ the set of s-times continuously differentiable functions on Ω. Additionally, if $\overline{\Omega}$ is compact, we set, for $f \in C^s(\Omega)$,

$$\|f\|_{C^s(\overline{\Omega})} := \max_{0\leq\|\boldsymbol{\alpha}\|_1\leq s} \sup_{x\in\Omega} |D^{\boldsymbol{\alpha}} f(x)| .$$

For any $s \in \mathbb{N}$ and $1 \leq p < \infty$, we define the Sobolev space $W^{s,p}(\Omega)$ by

$$W^{s,p}(\Omega) := \{f \in L^p(\Omega) : D^{\boldsymbol{\alpha}} f \in L^p(\Omega),\ \forall \boldsymbol{\alpha} \in \mathbb{N}^d \ with\ \|\boldsymbol{\alpha}\|_1 \leq s\} .$$

In particular, when $p = 2$, we define $H^s(\Omega) := W^{s,2}(\Omega)$ for any $s \in \mathbb{N}$. Moreover, for any $f \in W^{s,p}(\Omega)$ with $1 \leq p < \infty$, we define the Sobolev norm by

$$\|f\|_{W^{s,p}(\Omega)} := \left(\sum_{0\leq\|\boldsymbol{\alpha}\|_1\leq s} \|D^{\boldsymbol{\alpha}} f\|^p_{L^p(\Omega)} \right)^{1/p} .$$

When $p = \infty$, we set

$$\|f\|_{W^{s,\infty}(\Omega)} := \max_{0\leq\|\boldsymbol{\alpha}\|_1\leq s} \|D^{\boldsymbol{\alpha}} f\|_{L^\infty(\Omega)} .$$

We also introduce the Sobolev semi-norm by focusing on derivatives of exact order s. For $f \in W^{s,p}(\Omega)$ with $1 \leq p < \infty$, we define

$$|f|_{W^{s,p}(\Omega)} := \left(\sum_{\|\boldsymbol{\alpha}\|_1=s} \|D^{\boldsymbol{\alpha}} f\|^p_{L^p(\Omega)} \right)^{1/p},$$

and when $p = \infty$, we set

$$|f|_{W^{s,\infty}(\Omega)} := \max_{\|\boldsymbol{\alpha}\|_1=s} \|D^{\boldsymbol{\alpha}} f\|_{L^\infty(\Omega)} .$$

The notation specific to this paper are listed in Table 1.

Table 1: Table of Notation

Notation	Description
$\mathfrak{m}$	Total number of sub-networks in a parallel neural network.
M	Upper bound of the sum $\sum \lvert c_k \rvert$, where c_k are the linear coefficients combining the sub-networks.
W, L	The maximum width (number of neurons in the widest layer) and depth (number of layers) of each sub-network.
$B_{\boldsymbol{\theta}}$	Upper bound on the absolute values of sub-network weights and biases.
$\mathfrak{n}_\ell$	Total number of nonzero weights in the first ℓ layers of a sub-network.
$\mathfrak{D}(W, L, d)$	Total number of trainable weights in a sub-network with width W, depth L, and input dimension d.
$\boldsymbol{\theta}_k$	$(a^{(0)}_{k,1,1}, \ldots, a^{(L-1)}_{k,N_L,N_{L-1}}, b^{(0)}_{k,1}, \ldots, b^{(L-1)}_{k,N_L})$, weights of the k-th sub-network.
$\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}}$	$(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\mathfrak{m}})$, weights of all the $\mathfrak{m}$ sub-networks in the parallel neural network.
$\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}$	$(c_1, \ldots, c_{\mathfrak{m}})$, linear coefficients used to combine the $\mathfrak{m}$ sub-networks.
$\boldsymbol{\theta}^{\mathfrak{m}}_{\text{total}}$	$(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}})$, set of all weights in the parallel neural network.
B	Range of the uniform distribution $[-B, B]$ used to initialize the inner-layer weights and biases of each sub-network.
η	Projection radius for the ℓ_2-norm of the vector formed by the inner-layer parameters during projected gradient descent.
ζ	Projection radius for the ℓ_1-norm of the vector formed by the outer-layer parameters during projected gradient descent.
T	Number of iterations of the PGD algorithm.
λ	Learning rate of the PGD algorithm.
N_s	Number of training samples required to achieve the specified accuracy.
n	Regularity of the target solution u_0.
β	A scaling exponent determining how the projection radius η grows.

2.2 Topology of the Deep Networks

Let $L, d, N_0, \ldots, N_L \in \mathbb{N}$. We consider the function $\phi_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}$ that can be parameterized by a ρ-activated neural network of the form

$$\begin{aligned} &\phi_0(\boldsymbol{x}) = \boldsymbol{x}\,, \\ &\phi_\ell(\boldsymbol{x}) = \rho(\boldsymbol{A}_{\ell-1}\phi_{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}_{\ell-1}), \quad \ell = 1, \ldots, L-1\,, \\ &\phi_L(\boldsymbol{x}) = \boldsymbol{A}_{L-1}\phi_{L-1}(\boldsymbol{x}) + \boldsymbol{b}_{L-1}\,. \end{aligned} \tag{1}$$

where $\boldsymbol{A}_\ell = (a^{(\ell)}_{i,j}) \in \mathbb{R}^{N_{\ell+1} \times N_\ell}$, $\boldsymbol{b}_\ell = (b^{(\ell)}_i) \in \mathbb{R}^{N_{\ell+1}}$ with $N_0 = d$ and $N_L = 1$. The number $W := \max\{N_1, \ldots, N_L\}$ is called the width of the network, and L the depth. For

convenience, we denote $\mathfrak{n}_\ell$, $\ell = 1, \ldots, L$, as the number of nonzero weights in the first ℓ layers of the network, with $\mathfrak{n}_L \leq \mathfrak{D}(W, L, d)$. Here, $\mathfrak{D}(W, L, d)$ is defined as

$$\mathfrak{D}(W, L, d) := (W + 1)[(L - 2)W + d + 1]\,, \tag{2}$$

Meanwhile, W is generally greater than d in the following context. Therefore, we will also use the following estimate $\mathfrak{n}_L \leq \mathfrak{D}(W, L, d) \leq W(W + 1)L$ without loss of generality.

Let $\boldsymbol{\theta} = (a_{1,1}^{(0)}, \ldots, a_{N_L, N_{L-1}}^{(L-1)}, b_1^{(0)}, \ldots, b_{N_L}^{(L-1)})$ denote the weight vector of a neural network, and let Θ be the space of all such vectors. For a given activation function ρ, we define the function class $\mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ as the set of functions $\phi_{\boldsymbol{\theta}}$ realized by ρ-activated networks with width W, depth L, and parameters $\boldsymbol{\theta} \in \Theta$ constrained by $\|\boldsymbol{\theta}\|_\infty \leq B_{\boldsymbol{\theta}}$. Throughout this paper, we take the activation function to be $\rho = \tanh$.

Note that we can elevate any weight vector $\boldsymbol{\theta}$ to a $\mathfrak{D}(W, L, d)$-dimensional vector

$$\boldsymbol{\theta}' = (a_{1,1}^{(0)}, \ldots, a_{W,d}^{(0)}, a_{1,1}^{(1)}, \ldots, a_{W,W}^{(L-2)}, a_{1,1}^{(L-1)}, \ldots, a_{1,W}^{(L-1)}, b_1^{(0)}, \ldots, b_W^{(0)}, \ldots, b_W^{(L-2)}, b_1^{(L-1)})$$

by padding zeros, with $\phi_{\boldsymbol{\theta}}$ and $\phi_{\boldsymbol{\theta}'}$ representing the same element in $\mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$. Therefore, for any $\phi_{\boldsymbol{\theta}_1}, \phi_{\boldsymbol{\theta}_2} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$, we can align $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ to dimension $\mathfrak{D}(W, L, d)$, and perform vector operations on them. It also implies that we can formalize the parameter space Θ of $\mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ as

$$\Theta = [-B_{\boldsymbol{\theta}}, B_{\boldsymbol{\theta}}]^{\mathfrak{D}(W,L,d)}\,.$$

Further, we introduce a *Parallel Neural Network* class, $\mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}})$, which represents a linear combination of $\mathfrak{m}$ sub-networks. The structure is shown in Figure 1. Specifically, for any $\Phi_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$, it can be expressed as

$$\Phi_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k(\boldsymbol{x})\,, \quad c_k \in \mathbb{R}\,,$$

where $\phi_{\boldsymbol{\theta}}^k(\boldsymbol{x}) \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$.

Define $\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}} := (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\mathfrak{m}})$ where $\boldsymbol{\theta}_k = (a_{k,1,1}^{(0)}, \ldots, a_{k,N_L,N_{L-1}}^{(L-1)}, b_{k,1}^{(0)}, \ldots, b_{k,N_L}^{(L-1)})$. Define $\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}} := (c_1, \ldots, c_{\mathfrak{m}})$. Define $\Theta^{\mathfrak{m}}$ as the set of all weight vectors $\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m}} := (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}})$ that parameterize $\Phi_{\mathfrak{m},\boldsymbol{\theta}}$. Where it does not cause ambiguity, the symbol $\mathcal{PNN}$ will be used both as an abbreviation for some specific $\mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$, and to refer to a general parallel neural network class composed of multiple sub-networks.

2.3 Deep Ritz Method

Recall the deep Ritz method proposed in Weinan and Yu (2017). Let $[0, 1]^d$ be the unit hypercube on $\mathbb{R}^d$, Ω be a bounded open subset strictly contained in $[0, 1]^d$ and $\partial\Omega$ be the boundary of Ω. Consider the elliptic equation on Ω equipped with Neumann boundary condition:

$$-\Delta u + wu = h\,,\ \boldsymbol{x} \in \Omega\,; \qquad \frac{\partial u}{\partial \boldsymbol{n}} = g\,,\ \boldsymbol{x} \in \partial\Omega\,. \tag{3}$$

With the following assumptions on the known terms:

$$\partial\Omega \in C^{2+\alpha}\,, \quad h \in L^2(\Omega)\,, \quad g \in H^{1/2}(\partial\Omega)\,, \quad w \in C^\alpha(\bar{\Omega})\,, \quad w(x) \geq c_0 > 0\,,$$

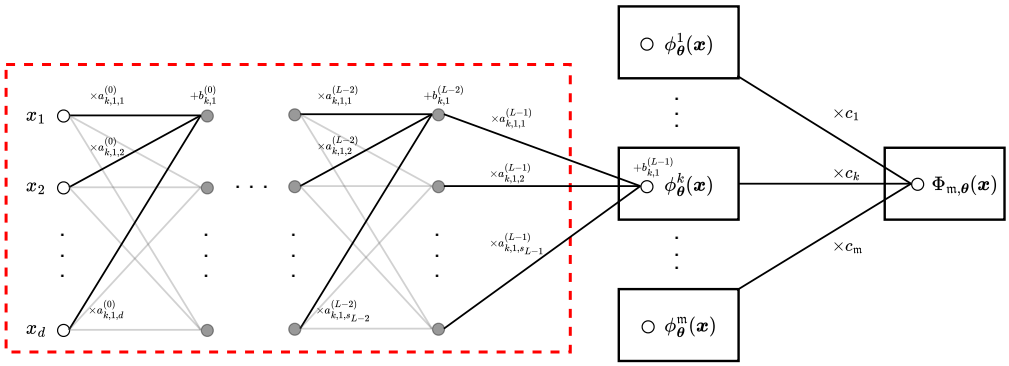

Figure 1: This figure illustrates the structure of the Parallel Neural Network. The structure within the red box represents the sub-neural networks, which are fully connected networks, where the dark-colored nodes signify activation functions.

where $0 < \alpha < 1$, Equation 3 has a unique weak solution $u_0 \in H^2(\Omega)$ (Agmon et al., 1959). We further assume that $\|u_0\|_{H^1(\Omega)} \leq 1$. Let $B_0 = \max\{\|h\|_{L^\infty(\Omega)}, \|g\|_{L^\infty(\partial\Omega)}, \|w\|_{L^\infty(\Omega)}\}$. Define the energy functional $\mathcal{L}$ as follows:

$$\mathcal{L}(u) = \int_\Omega \left(\frac{\|\nabla u\|_2^2 + w|u|^2}{2} - hu \right) \mathrm{d}x - \int_{\partial\Omega} gTu \, \mathrm{d}s \,, \tag{4}$$

where T is the trace operator. Proposition 1 demonstrates that minimizing $\mathcal{L}(u)$ in Equation 4 is equivalent to reducing the distance between u and u_0 in the H^1 norm.

Proposition 1 *For any $u \in H^1(\Omega)$, it holds that*

$$\frac{c_0 \wedge 1}{2} \|u - u_0\|_{H^1(\Omega)}^2 \leq \mathcal{L}(u) - \mathcal{L}(u_0) \leq \frac{B_0 \vee 1}{2} \|u - u_0\|_{H^1(\Omega)}^2 \,.$$

Proof For any $u \in H^1(\Omega)$, set $v = u - u_0$. Then, it holds that

$$\begin{aligned}
\mathcal{L}(u) &= \mathcal{L}(u_0 + v) \\
&= \int_\Omega \frac{\|\nabla(u_0 + v)\|_2^2 + w|u_0 + v|^2}{2} \mathrm{d}x - \int_\Omega h(u_0 + v) \mathrm{d}x - \int_{\partial\Omega} g(Tu_0 + Tv) \mathrm{d}s \\
&= \int_\Omega \frac{\|\nabla u_0\|_2^2 + w|u_0|^2}{2} \mathrm{d}x - \int_\Omega hu_0 \mathrm{d}x - \int_{\partial\Omega} gTu_0 \, \mathrm{d}s + \int_\Omega \frac{\|\nabla v\|_2^2 + w|v|^2}{2} \mathrm{d}x \\
&\quad + \int_\Omega \langle \nabla u_0, \nabla v \rangle \, \mathrm{d}x + \int_\Omega wu_0 v \, \mathrm{d}x - \int_\Omega hv \, \mathrm{d}x - \int_{\partial\Omega} gTv \, \mathrm{d}s \\
&= \mathcal{L}(u_0) + \int_\Omega \frac{\|\nabla v\|_2^2 + w|v|^2}{2} \mathrm{d}x \,,
\end{aligned}$$

where the last equality is due to the fact that u_0 is the weak solution of Equation 3. Hence

$$\frac{c_0 \wedge 1}{2} \|v\|_{H^1(\Omega)}^2 \leq \mathcal{L}(u) - \mathcal{L}(u_0) = \int_\Omega \frac{\|\nabla v\|_2^2 + w|v|^2}{2} \mathrm{d}x \leq \frac{\|w\|_{L^\infty(\Omega)} \vee 1}{2} \|v\|_{H^1(\Omega)}^2 \,,$$

that is,

$$\frac{c_0 \wedge 1}{2}\|u-u_0\|^2_{H^1(\Omega)} \leq \mathcal{L}(u)-\mathcal{L}(u_0) \leq \frac{B_0 \vee 1}{2}\|u-u_0\|^2_{H^1(\Omega)}\,.$$

This concludes the proof. ■

To facilitate implementation of deep learning algorithms, we use Monte Carlo method to discretize the integral type functional $\mathcal{L}$. First, Equation 4 is rewritten as

$$\mathcal{L}(u) = |\Omega| \mathop{\mathbb{E}}_{X\sim U(\Omega)}\left[\frac{\|\nabla u(X)\|_2^2}{2}+\frac{w(X)u^2(X)}{2}-u(X)h(X)\right]-|\partial\Omega|\mathop{\mathbb{E}}_{Y\sim U(\partial\Omega)}[Tu(Y)g(Y)]\,, \quad (5)$$

where $U(\Omega)$, $U(\partial\Omega)$ are the uniform distribution on Ω and $\partial\Omega$. Based on Equation 5, we introduce the discrete version $\widehat{\mathcal{L}}(u)$:

$$\widehat{\mathcal{L}}(u) = \frac{|\Omega|}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\left[\frac{\|\nabla u(X_p)\|_2^2}{2}+\frac{w(X_p)u^2(X_p)}{2}-u(X_p)h(X_p)\right]-\frac{|\partial\Omega|}{N_b}\sum_{q=1}^{N_b}u(Y_q)g(Y_q)\,, \quad (6)$$

where $\{X_p\}_{p=1}^{N_{\text{in}}} \sim_{\text{i.i.d.}} U(\Omega)$, $\{Y_q\}_{q=1}^{N_b} \sim_{\text{i.i.d.}} U(\partial\Omega)$. Then, we select a deep neural network class $\mathcal{F}_{\boldsymbol{\theta}}$, within which we will minimize $\widehat{\mathcal{L}}(u_{\boldsymbol{\theta}})$ for $u_{\boldsymbol{\theta}} \in \mathcal{F}_{\boldsymbol{\theta}}$.

In this paper, our choice is the $\mathcal{PNN}$ parallel neural network $u_{\mathfrak{m},\boldsymbol{\theta}} = \sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k$ introduced in Section 2.2, which comprises $\mathfrak{m}$ sub-networks. Now, Equation 6 turns into:

$$\begin{aligned}\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}}) = &\frac{|\Omega|}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\left[\frac{\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)\|_2^2}{2}+\frac{w(X_p)u^2_{\mathfrak{m},\boldsymbol{\theta}}(X_p)}{2}-u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)h(X_p)\right]\\ &-\frac{|\partial\Omega|}{N_b}\sum_{q=1}^{N_b}u_{\mathfrak{m},\boldsymbol{\theta}}(Y_q)g(Y_q)\,. \end{aligned} \quad (7)$$

2.4 Projected Gradient Descent

We use the PGD algorithm to minimize $\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})$ in Equation 7, which is an iterative optimization method suitable for constrained optimization problems.

Since the Monte Carlo samples $\{X_p\}_{p=1}^{N_{\text{in}}}$, $\{Y_q\}_{q=1}^{N_b}$ are fixed during the optimization process, $\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})$ becomes a function solely dependent on the weights $\boldsymbol{\theta}^{\mathfrak{m}}_{\text{total}}$, and we denote it as $\widehat{F}(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{total}}) = \widehat{F}(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}})$. The PGD algorithm consists of the following three steps:

Initialization. We start with an initial guess $(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{total}})^{[0]} = (\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}})^{[0]}$ as follows:

(i) For the linear coefficients $\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}$, set

$$(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}})^{[0]} = \mathbf{0}\,, \quad i.e., \quad (c_k)^{[0]} = 0 \quad (k=1,\ldots,\mathfrak{m})\,. \quad (8)$$

(ii) For the sub-network parameters $\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}}$, initialize each element in $(\boldsymbol{\theta}^{\mathfrak{m}}_{\text{in}})^{[0]}$ to follow the same uniform distribution $U[-B,B]$ independently, that is,

$$\left(a^{(\ell)}_{k,i,j}\right)^{[0]} \sim_{\text{i.i.d.}} U[-B,B]\,, \quad \left(b^{(\ell)}_{k,i}\right)^{[0]} \sim_{\text{i.i.d.}} U[-B,B]\,. \quad (9)$$

Constraint Set. Then, we choose $\eta, \zeta > 0$, and determine the constraint set as follows:

(i) Let A_η be the (random) set of all weight vectors $\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}$ which satisfy

$$\left\|\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}} - (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\right\|_2 \leq \eta\,. \tag{10}$$

(ii) Let B_ζ be the set of all weight vectors $\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}$ which satisfy

$$\left\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}\right\|_1 = \sum_{k=1}^{\mathfrak{m}} |c_k| \leq \zeta\,. \tag{11}$$

Iterative Update. Finally, let $T \in \mathbb{N}^+$, $\lambda > 0$. For each iteration $t = 0, \ldots, T-1$, do

(i) Compute the gradient of the objective function at the current point:

$$\boldsymbol{g}^{[t]} = \nabla_{\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m}}} \widehat{F}\big(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},[t]}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},[t]}\big)\,.$$

(ii) Update the weight vector by first performing a gradient descent step with step size λ and then projecting the result onto the feasible set:

$$\big(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}\big)^{[t+1]} = \text{Proj}_{A_\eta \times B_\zeta}\big\{(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}})^{[t]} - \lambda\, \boldsymbol{g}^{[t]}\big\}\,, \tag{12}$$

where $\text{Proj}_{\mathcal{C}}$ denotes the projection operator in the Euclidean space.

Remark 1 *The projection onto the ℓ_2 ball A_η can be computed in closed form, while the projection onto the ℓ_1 ball B_ζ can be implemented with computational complexity that scales linearly with the dimension (Duchi et al., 2008). When $N_{\text{in}} = N_b = N_s$, a single iteration of our PGD algorithm requires $\mathcal{O}(\mathfrak{m}W^2L^2N_s)$ operations, as demonstrated in Appendix D.*

Remark 2 *The projection step plays a crucial role in the subsequent error analysis: it constrains the range of variation of the neural network parameters and potentially limits the complexity of the neural network class. This makes it possible to derive size-independent statistical error by bounding the Rademacher complexity of the potentially highly overparameterized parallel neural network class.*

In the following, we will use $\mathcal{A}$ to represent the PGD algorithm, and use $u_{\mathcal{A}}$ to denote the output of $\mathcal{A}$ which serves as the final solution. It is evident that $u_{\mathcal{A}}$ is exactly $u_{\mathfrak{m},\boldsymbol{\theta}}$ parameterized with $(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}})^{[T]}$.

2.5 Main Result

We now present the main theorem of this work, which provides a comprehensive end-to-end error analysis for solving elliptic equations via the deep Ritz method in the over-parameterized setting. The complete proof is given in Section 3.5.

Theorem 1 *Suppose that $u_0 \in H^n(\Omega)$ for $n \geq 2$ is the solution of Equation 3. We consider the application of the deep Ritz method to solve Equation 3 using a parallel neural network ($\mathcal{PNN}$) architecture comprising $\mathfrak{m}$ parallel sub-networks, each with width W and depth L. The network parameters are initialized according to Equations 8 and 9, with linear coefficients connecting the sub-networks set to $\mathbf{0}$, and each sub-network weight drawn independently from the uniform distribution $U[-B, B]$. Let η and ζ be the projection radius described in Equations 10 and 11, respectively. Let $N_{\text{in}} = N_b = N_s$ denote the Monte Carlo sample size in Equation 7, and let $u_{\mathcal{A}}$ be the output of the PGD algorithm in Equation 12 with T iterations and step size λ. For any $0 < \epsilon \ll 1$, set*

$$\begin{aligned}
&\mathfrak{m} = \lceil C\epsilon^{-\tilde{C}_1(\mu,d,\beta_0,n)} \rceil\,, \qquad W = 2^{\lceil \log_2(d+n-1)\rceil+1}\,, \qquad L = \lceil \log_2(d+n-1)\rceil + 2\,,\\
&B = C\epsilon^{-\frac{2d+2n}{n-\mu-1}}\,, \qquad \eta = \epsilon^{-\beta}\,, \qquad \zeta = C\epsilon^{-\frac{3d}{2(n-\mu-1)}}\,,\\
&T = \lceil C\epsilon^{-\tilde{C}_2(\mu,d,\beta_0,n)} \rceil\,, \qquad \lambda = C\epsilon^{\tilde{C}_2(\mu,d,\beta_0,n)}\,, \qquad N_s = \lceil C\epsilon^{-\tilde{C}_3(\mu,d,\beta_0,n)} \rceil\,.
\end{aligned}$$

Then, with probability at least $1 - 2\epsilon^{\tilde{C}_3(\mu,d,\beta_0,n)}$, the total error satisfies

$$\|u_{\mathcal{A}} - u_0\|^2_{H^1(\Omega)} \leq C\epsilon \log^{1/2}(C\epsilon^{-1}) = \tilde{\mathcal{O}}(\epsilon)\,,$$

where

$$\begin{aligned}
&\beta > 0\,, \qquad \beta_0 = \max\{\beta,\, (2d+2n)(n-\mu-1)^{-1}\}\,,\\
&\tilde{C}_1(\mu,d,\beta_0,n) = \frac{C'(d+n)^3\log_2^2(d+n-1)}{n-\mu-1} + 11\beta_0\log_2(d+n-1) + 36\beta_0\,,\\
&\tilde{C}_2(\mu,d,\beta_0,n) = \frac{C''(d+n)^3\log_2^2(d+n-1)}{n-\mu-1} + 15\beta_0\log_2(d+n-1) + 48\beta_0\,,\\
&\tilde{C}_3(\mu,d,\beta_0,n) = 4\beta_0\log_2(d+n-1) + 15\beta_0\,.
\end{aligned}$$

Here, C denotes a place-by-place defined constant depending only on Ω, W, L, d, B_0 and n; C' and C'' are positive constants; and μ is an arbitrarily small positive constant.

Remark 3 *The assumption $u_0 \in H^n(\Omega)$ for $n \geq 2$ can be achieved by increasing the regularity of the coefficient w and the right-hand side functions h, g in Equation 3. For instance, with $\partial\Omega \in C^n$, such assumption would be realized if we have $h \in H^{n-2}(\Omega)$, $g \in H^{n-3/2}(\partial\Omega)$ and $w \in C^{n-2}(\bar{\Omega})$. See Agmon et al. (1959) for proof.*

Remark 4 *Note that in Theorem 1, when d and n are fixed, the architecture of each sub-network (width W and depth L) remains constant. Consequently, the number of parallel sub-networks $\mathfrak{m}$ serves as the primary measure of over-parameterization in our analysis.*

Remark 5 *A notable feature of our analysis is that it does not require the neural network parameters to remain in close proximity to their initialization values during training—a restrictive requirement commonly imposed in prior optimization error analyses (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou and Gu, 2019; Liu et al., 2022; Chizat et al., 2019; Nguyen, 2021; Lu et al., 2020; Mahankali et al., 2024). However, when the projection radius η grows faster than the initialization range B ($\beta_0 = \beta$), achieving the same precision potentially demands greater computational resources: increased Monte Carlo samples, higher levels of over-parameterization, more iterations, and smaller step sizes.*

Remark 6 *From Theorem 1, achieving the error bound $\tilde{\mathcal{O}}(\epsilon)$ requires a sample size of*

$$N_s = \lceil C\epsilon^{-\tilde{C}_3(\mu,d,\beta_0,n)} \rceil,$$

where $\tilde{C}_3(\mu,d,\beta_0,n) = 4\beta_0\log_2(d+n-1)+15\beta_0$ with $\beta_0 = \max\{\beta,(2d+2n)(n-\mu-1)^{-1}\}$. This yields the convergence rate

$$\|u_{\mathcal{A}} - u_0\|^2_{H^1(\Omega)} \leq \tilde{\mathcal{O}}\left(N_s^{-\frac{n-\mu-1}{[8\log_2(d+n-1)+30](d+n)}}\right),$$

if we let $\beta_0 = (2d+2n)(n-\mu-1)^{-1}$. For comparison, Lu et al. (2021c) established a rate of $N_s^{-\frac{2n-2}{d+2n-4}}$, which achieves the minimax lower bound. While our current result does not attain this optimal rate, improving the convergence analysis remains an active direction for our future work.

3. Proofs

In this section, we outline the five key steps required to prove Theorem 1. The full details of the proof are provided in the Appendix.

Step 1: Decomposing the total error. The total error between $u_{\mathcal{A}}$ and u_0 can be decomposed into three main components: approximation error, statistical error, and a novel and tighter optimization error; details can be found in Section 3.1.

Step 2: Constructing a parallel neural network with explicit weight bound to approximate in the Sobolev space. Building on methods from Yarotsky (2017), Jiao et al. (2023e), and Gühring and Raslan (2021), we derive an approximation error bound for neural networks in the Sobolev space. This bound is achieved by explicitly constructing tanh-activated neural networks that approximate local Taylor polynomials. Notably, the constructed network has a parallel architecture, meaning the final neural network is a linear combination of many structurally identical fully connected sub-networks, see Section 2.2 for detail. By construction, we explicitly control the weight bound of the deep network, which is critical for the statistical error analysis in the over-parameterized setting, as demonstrated in Step 4. We also use this constructed over-parameterized network to define and analyze the novel optimization error term in Step 3.

Step 3: Using the property of over-parameterization to analyze the new optimization error. As demonstrated in Equation 22, the optimization error is bounded by the sum of 'initialization error' and 'iteration error'. Through over-parameterization, the neural network's initialization captures sufficient information about the optimal approximation network constructed in Step 2, enabling effective control of the initialization error. Concurrently, the iteration error of the PGD algorithm is controlled by selecting sufficiently large iteration counts and appropriately small step sizes. A distinguishing feature of our optimization error analysis is that the projection radius for the inner parameters of parallel sub-networks can diverge at a controlled rate, thereby circumventing the training stagnation phenomenon encountered in previous optimization analyses.

Step 4: Analyzing the statistical error for over-parameterized deep neural network class. The PGD optimization in Step 3 produces an output $u_{\mathcal{A}}$ within an over-parameterized neural network class. Standard tools from empirical process theory (Van

Der Vaart and Wellner, 1996; van de Geer, 2000; Giné and Nickl, 2021) cannot be directly applied to bound the statistical error in this context, as they would yield upper bounds that grow uncontrollably in the over-parameterized regime. Instead, we leverage the explicit weight constraints established in Step 2 and the projection operations in Step 3 to derive size-independent statistical error bounds, which fully exploit the parallel architecture of our neural network framework. By appropriately calibrating the Monte Carlo sample sizes for interior and boundary points, we ensure the statistical error remains bounded within acceptable limits. See Section 3.4 for details.

Step 5: Synthesizing the error analysis from each component. By synthesizing the analysis of approximation, optimization, and statistical error from Step 2 to Step 4, we could control the total error between $u_{\mathcal{A}}$ and u_0 within the desired precision under appropriate parameter settings, thus proving our main result Theorem 1.

3.1 New Error Decomposition

To perform an end-to-end error analysis between $u_{\mathcal{A}}$ and u_0, we establish the following error decomposition theorem. A key innovation enabling this analysis is the introduction of a novel 'optimization error', defined as

$$\mathcal{E}_{\text{opt}}^{-} := \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u}),$$

where $\bar{u}$ is the best approximation to u_0 in a parallel neural network class $\mathcal{PNN}'$ that may differ from the algorithm's $\mathcal{PNN}$ class, formally defined as

$$\bar{u} \in \operatorname*{argmin}_{u \in \mathcal{PNN}'} \|u - u_0\|_{H^1(\Omega)}^2. \tag{13}$$

Theorem 2 *Let $u_{\mathcal{A}}$ be the PGD algorithm output when using DRM to solve Equation 3 and $\bar{u} \in \mathcal{PNN}'$ defined in Equation 13, the H^1 distance between $u_{\mathcal{A}}$ and the true solution u_0 can be decomposed into*

$$\begin{aligned}
&\|u_{\mathcal{A}} - u_0\|_{H^1(\Omega)}^2 \\
&\quad \le \frac{2}{c_0 \wedge 1} \Bigg\{ \underbrace{\frac{B_0 \vee 1}{2} \|\bar{u} - u_0\|_{H^1(\Omega)}^2}_{\mathcal{E}_{app}} + \underbrace{\Big[\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})\Big]}_{\mathcal{E}_{opt}^{-}} + \underbrace{2 \sup_{u \in \mathcal{PNN}} \big|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)\big|}_{\mathcal{E}_{sta}} \Bigg\}.
\end{aligned}$$

Proof By Proposition 1, it holds that

$$\begin{aligned}
&\|u_{\mathcal{A}} - u_0\|_{H^1(\Omega)}^2 \\
&\quad \le \frac{2}{c_0 \wedge 1} \Big\{ \big[\mathcal{L}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\mathcal{A}})\big] + \big[\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})\big] + \big[\widehat{\mathcal{L}}(\bar{u}) - \mathcal{L}(\bar{u})\big] + \big[\mathcal{L}(\bar{u}) - \mathcal{L}(u_0)\big] \Big\} \\
&\quad \le \frac{2}{c_0 \wedge 1} \Big\{ \big[\mathcal{L}(\bar{u}) - \mathcal{L}(u_0)\big] + \big[\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})\big] + 2 \sup_{u \in \mathcal{PNN}} \big|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)\big| \Big\} \\
&\quad \le \frac{2}{c_0 \wedge 1} \Bigg\{ \underbrace{\frac{B_0 \vee 1}{2} \|\bar{u} - u_0\|_{H^1(\Omega)}^2}_{\mathcal{E}_{app}} + \underbrace{\Big[\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})\Big]}_{\mathcal{E}_{opt}^{-}} + \underbrace{2 \sup_{u \in \mathcal{PNN}} \big|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)\big|}_{\mathcal{E}_{sta}} \Bigg\}. \quad \blacksquare
\end{aligned}$$

Remark 7 *The optimization error is traditionally defined as* $\mathcal{E}_{opt} = \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\hat{u})$*, where* $\hat{u}$ *denotes the Empirical Risk Minimization (ERM) estimator of Equation 7:*

$$\hat{u} = \underset{u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}}{\operatorname{argmin}} \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}}) \,.$$

Since we have

$$\mathcal{E}_{opt}^{-} = \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\hat{u}) + \widehat{\mathcal{L}}(\hat{u}) - \widehat{\mathcal{L}}(\bar{u}) \leq \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\hat{u}) + 0 = \mathcal{E}_{opt} \,,$$

the newly defined $\mathcal{E}_{opt}^{-}$ *is clearly tighter than* $\mathcal{E}_{opt}$*. Additionally, due to the highly non-convex training objective of deep neural networks, it is challenging to obtain detailed information about* $\hat{u}$*, making analysis of* $\mathcal{E}_{opt}$ *difficult. In contrast, the best approximation element* $\bar{u}$ *is explicitly constructed according to the target solution, as shown in Equation 13, so its information can be fully grasped, greatly facilitating the analysis of* $\mathcal{E}_{opt}^{-}$*. In Section 3.3, we will prove that when the over-parameterization level is sufficiently high, the initialization parameters of neural networks will contain sufficient information about* $\bar{u}$ *with high probability, and meanwhile, the iteration error of the PGD algorithm can be well controlled. Combining these two points, we can control* $\mathcal{E}_{opt}^{-}$ *with arbitrary precision, thus achieving a complete end-to-end error analysis between* $u_{\mathcal{A}}$ *and* u_0*.*

3.2 Approximation Error

In this section, we derive an upper bound on the approximation error, which reflects the ability of the constructed neural network $\bar{u}$ to approximate the true solution u_0. Recall that

$$\mathcal{E}_{app} := \|\bar{u} - u_0\|_{H^1(\Omega)}^2 \,, \tag{14}$$

where $\bar{u}$ is defined in Equation 13.

Leveraging insights from Gühring and Raslan (2021), we establish that for any prescribed accuracy $\epsilon > 0$, each function $f \in \mathcal{F}_{n,d,p}$ admits an ϵ-approximation in Sobolev norm $W^{1,p}$ (where $n, p \in \mathbb{N}$, $n \geq 2$, and $1 \leq p \leq \infty$) within a tanh-activated parallel neural network class $\mathcal{PNN}'$. The function class $\mathcal{F}_{n,d,p}$ is defined as

$$\mathcal{F}_{n,d,p} := \left\{ f \in W^{n,p}(\Omega) : \|f\|_{W^{n,p}(\Omega)} \leq 1 \right\}.$$

A complete proof is provided in Appendix A. By setting $p = 2$ in Theorem 7 (Appendix A), we obtain the following approximation result, which precisely aligns with our requirements.

Theorem 3 *Given any* $u_0 \in \mathcal{F}_{n,d,2}$*, for some sufficiently small* $\epsilon^* > 0$ *and any* $0 < \epsilon < \epsilon^*$*, there exists a neural network* $\bar{u} = u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ *with*

$$\bar{\mathfrak{m}} = \lceil C_1 \epsilon^{-\frac{d}{n-\mu-1}} \rceil \,, \quad \bar{M} = C_2 \epsilon^{-\frac{3d}{2(n-\mu-1)}} \,, \quad \bar{W} = 2^{\lceil \log_2(d+n-1) \rceil + 1} \,,$$
$$\bar{L} = \lceil \log_2(d+n-1) \rceil + 2 \,, \quad B_{\bar{\boldsymbol{\theta}}} = C_3 \epsilon^{-\frac{2d+2n}{n-\mu-1}} \,,$$

such that

$$\|u_0 - u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\|_{H^1(\Omega)} \leq \epsilon \,,$$

where $n \geq 2$ *and* $\mu > 0$ *is an arbitrarily small positive number;* C_1, C_2 *and* C_3 *are universal constants which only depend on* d *and* n*.*

3.3 Optimization Error

Now, we analyze the optimization error $\mathcal{E}_{opt}^{-}$, defined for the PGD algorithm output $u_{\mathcal{A}} \in \mathcal{PNN}$ (Section 2.4) and the best approximation element $\bar{u} \in \mathcal{PNN}'$ (Equation 13) as

$$\mathcal{E}_{opt}^{-} := \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})\,.$$

We choose $\bar{u} = u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ in Theorem 3:

$$\mathcal{E}_{opt}^{-} = \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}})\,. \tag{15}$$

Intuitively, the weights of $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$ serve as 'target parameters' during the optimization process, specifically the sub-network parameters $(\bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_{\bar{\mathfrak{m}}})$ and the linear coefficients $(\bar{c}_1, \ldots, \bar{c}_{\bar{\mathfrak{m}}})$. Accordingly, we configure the implemented network $u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}$ with sub-network width $W = \bar{W}$, depth $L = \bar{L}$, and uniform distribution range $B = B_{\bar{\boldsymbol{\theta}}}$ in Equation 9, which aligns with the parameter specifications of $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$. Given random sub-network initialization $(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\mathfrak{m}})^{[0]}$, we seek to characterize the set of 'sufficiently good' initializations relative to $(\bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_{\bar{\mathfrak{m}}})$. We formalize this notion in the following definition.

Definition 1 *Let $G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$ denote the event where for each target parameter $\bar{\boldsymbol{\theta}}_k$, $k = 1, \ldots, \bar{\mathfrak{m}}$, there exist at least R distinct sub-network weight vectors $(\boldsymbol{\theta}_{i_{k,v}})^{[0]}$, $v = 1, \ldots, R$, in the random initialization $(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}$ such that*

$$\|(\boldsymbol{\theta}_{i_{k,v}})^{[0]} - \bar{\boldsymbol{\theta}}_k\|_\infty \leq \delta\,, \quad k = 1, \ldots, \bar{\mathfrak{m}}\,, \quad v = 1, \ldots, R\,.$$

Furthermore, we require that $i_{k,v} \neq i_{k',v'}$ whenever $k \neq k'$ or $v \neq v'$.

Remark 8 *Simply put, $G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$ ensures that for each target $\bar{\boldsymbol{\theta}}_k$, at least R sub-networks have already sufficiently approximated it during parameter initialization phase.*

In the subsequent analysis, we set the number of sub-networks $\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q$, where $R, Q \in \mathbb{N}$. To formalize the random indices $i_{k,v}$ from Definition 1, we introduce integer-valued random variables:

$$s_{k,v}(\omega)\,, \quad k = 1, \ldots, \bar{\mathfrak{m}}\,, \quad v = 1, \ldots, R\,. \tag{16}$$

Assuming the $\mathfrak{m}$ sub-networks are arranged in a predetermined order, we define these variables as follows: For $\omega \in G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$, when $k = 1$, we let $s_{1,v} \in [\mathfrak{m}]$ denote the index of the v-th sub-network satisfying $\|(\boldsymbol{\theta}_\cdot)^{[0]} - \bar{\boldsymbol{\theta}}_1\|_\infty \leq \delta$; when $k > 1$, we let $s_{k,v} \in [\mathfrak{m}] \setminus \{s_{l,v} : l < k, v \leq R\}$ denote the index of the v-th sub-network among the remaining $\mathfrak{m} - (k-1)R$ sub-networks satisfying $\|(\boldsymbol{\theta}_\cdot)^{[0]} - \bar{\boldsymbol{\theta}}_k\|_\infty \leq \delta$. For $\omega \notin G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$, we set $s_{k,v} = (k-1)R + v$. Intuitively, $s_{k,v}$ picks out the 'good' initialization weight vectors.

Based on $s_{k,v}$ and $(\bar{c}_1, \ldots, \bar{c}_{\bar{\mathfrak{m}}})$, we define a set of 'transition parameters' to bridge $u_{\mathcal{A}}$ and the target $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$:

$$\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*} := (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},*}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*})\,, \quad \text{where } \boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},*} := (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\,. \tag{17}$$

For the linear coefficients $\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*} := (c_1^*, \ldots, c_{\mathfrak{m}}^*)$, we define:

$$c_{s_{k,v}}^* := \frac{\bar{c}_k}{R}\,, \quad k = 1, \ldots, \bar{\mathfrak{m}}\,, \quad v = 1, \ldots, R\,, \tag{18}$$

$$c_q^* := 0\,, \quad q \notin \{s_{k,v} : k = 1, \ldots, \bar{\mathfrak{m}}\,,\ v = 1, \ldots, R\}\,. \tag{19}$$

When $u_{\mathfrak{m},\boldsymbol{\theta}}$ is parameterized with $\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*}$, we denote it as $u_{\mathfrak{m}}^*$ and expand it as:

$$u_{\mathfrak{m}}^*(\boldsymbol{x}) = \sum_{s=1}^{\mathfrak{m}} c_s^* \cdot \left(\phi_{\boldsymbol{\theta}}^s\right)^{[0]}(\boldsymbol{x}) = \sum_{k=1}^{\bar{\mathfrak{m}}} \sum_{v=1}^{R} \frac{\bar{c}_k}{R} \cdot \left(\phi_{\boldsymbol{\theta}}^{s_{k,v}}\right)^{[0]}(\boldsymbol{x})\,, \tag{20}$$

where $(\phi_{\boldsymbol{\theta}}^s)^{[0]}$ represents the s-th sub-network parameterized by $(\boldsymbol{\theta}_s)^{[0]}$. To clarify the relationship between $u_{\mathfrak{m}}^*$ and $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$, we express the latter as:

$$u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(\boldsymbol{x}) = \sum_{k=1}^{\bar{\mathfrak{m}}} \bar{c}_k \cdot \phi_{\bar{\boldsymbol{\theta}}}^k(\boldsymbol{x}) = \sum_{k=1}^{\bar{\mathfrak{m}}} \sum_{v=1}^{R} \frac{\bar{c}_k}{R} \cdot \phi_{\bar{\boldsymbol{\theta}}}^k(\boldsymbol{x})\,, \tag{21}$$

where $\phi_{\bar{\boldsymbol{\theta}}}^k$ is the k-th sub-network in $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$ with weights $\bar{\boldsymbol{\theta}}_k$. When $\bar{\boldsymbol{\theta}}_k$ and $(\boldsymbol{\theta}_{s_{k,v}})^{[0]}$ are sufficiently close—a condition ensured by event $G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$—we can expect $u_{\mathfrak{m}}^*$ and $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$ to be similarly proximate.

Using $u_{\mathfrak{m}}^*$ as an intermediate term, we decompose the optimization error $\mathcal{E}_{opt}^-$ into:

$$\mathcal{E}_{opt}^- = \widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}) = \underbrace{\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m}}^*)}_{\text{iteration error}} + \underbrace{\widehat{\mathcal{L}}(u_{\mathfrak{m}}^*) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}})}_{\text{initialization error}}\,. \tag{22}$$

Iteration Error: Recall that $u_{\mathcal{A}}$ is exactly $u_{\mathfrak{m},\boldsymbol{\theta}}$ parameterized with $(\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m}})^{[T]}$, the final output of iterative Equation 12, and $\widehat{\mathcal{L}}(u_{\mathcal{A}})$ can be denoted as $\widehat{F}(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},[T]}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},[T]})$. Similarly, $\widehat{\mathcal{L}}(u_{\mathfrak{m}}^*)$ can be expressed as $\widehat{F}(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},[0]}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*})$. Hence, the iteration error measures how 'close' the PGD algorithm output $(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}})^{[T]}$ is to the transition parameters $\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*} = (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m},[0]}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*})$.

We now characterize the specific $\mathcal{PNN}$ class containing $u_{\mathcal{A}}$. By Equation 11, the linear coefficients satisfy $\|(\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}})^{[t]}\|_1 \le \zeta$ during iteration. For sub-network parameters, we have

$$\|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[t]}\|_\infty \le \|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\|_\infty + \|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[t]} - (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\|_\infty \le \|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\|_\infty + \|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[t]} - (\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}\|_2\,,$$

which, by Equation 10, implies $\|(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[t]}\|_\infty \le B_{\bar{\boldsymbol{\theta}}} + \eta$. Consequently,

$$u_{\mathcal{A}} \in \mathcal{PNN}(\mathfrak{m}, \zeta, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})\,.$$

Proposition 2 establishes that by setting $\zeta = \bar{M}$ in Equation 11 and appropriately choosing the parameter R, iteration count T, and step size λ, we can bound the iteration error to any desired precision. This is achieved by using the strong convexity of the loss function $\widehat{F}(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}) := \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})$ with respect to the linear coefficients $\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}$ and its Lipschitz continuity with respect to the sub-network parameters $\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}$, as detailed in Appendix C.1.

Proposition 2 *Let $\zeta = \bar{M}$ in Equation 11. Then, the output $u_{\mathcal{A}}$ of the PGD algorithm belongs to $\mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$. When we run the algorithm with step size*

$$\lambda = \min\{T^{-1}, 2C_4^{-1}\mathfrak{m}^{-1}\bar{M}^{-2}(B_{\bar{\boldsymbol{\theta}}} + \eta)^{-4\bar{L}}\}\,,$$

where T is the total number of iterations and η is the projection radius in Equation 10, the iteration error in Equation 22 satisfies

$$\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m}}^*) \leq \frac{\bar{M}^2}{2R} + \frac{C_5 \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{3\bar{L}} \eta}{\sqrt{R}} + \frac{C_6 \mathfrak{m} \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{T}\,.$$

Here, $u_{\mathfrak{m}}^* \in \mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ is defined in Equation 20, and C_4, C_5, C_6 are universal constants depending only on $\Omega, \bar{W}, \bar{L}, d$ and B_0.

Initialization Error: Next, we analyze the initialization error. By Equations 20 and 21, this term can be effectively controlled when the target weights $\bar{\boldsymbol{\theta}}_k$ and their corresponding random initializations $(\boldsymbol{\theta}_{s_{k,v}})^{[0]}$ are sufficiently close. To bound this error precisely, we require $\mathbb{P}(G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta})$ from Definition 1 to approach 1 as $\delta \to 0$, which inherently constrains the minimum value of $\mathfrak{m}$.

Proposition 3 formalizes this relationship, demonstrating that by appropriately scaling the number of sub-networks in $u_{\mathfrak{m},\boldsymbol{\theta}}$, we can bound the initialization error with arbitrarily high probability and precision. The proof is presented in Appendix C.2.

Proposition 3 *Consider $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ from Theorem 3. For any $\delta > 0$ and $R, Q \in \mathbb{N}$ with Q sufficiently large, if we set $\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q$, then with probability at least*

$$1 - \bar{\mathfrak{m}} R \Big[1 - \delta^{\bar{W}(\bar{W}+1)\bar{L}} (2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^Q\,,$$

the initialization error in Equation 22 satisfies

$$\widehat{\mathcal{L}}(u_{\mathfrak{m}}^*) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}) \leq C_7 \bar{M}^2 B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}} \delta\,,$$

where $u_{\mathfrak{m}}^*$ is defined in Equation 20 and C_7 is a universal constant depending only on $\Omega, \bar{W}, \bar{L}, d$ and B_0.

Combining Propositions 2 and 3, we obtain the following estimate of $\mathcal{E}_{opt}^-$.

Theorem 4 *Consider $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ from Theorem 3. For any $\delta > 0$ and $R, Q \in \mathbb{N}$ with Q sufficiently large, set $\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q$. Let $u_{\mathcal{A}} \in \mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$ denote the output of the PGD algorithm in Equation 12 with $\zeta = \bar{M}$ in Equation 11 and step size*

$$\lambda = \min\{T^{-1}, 2C_4^{-1} \bar{\mathfrak{m}}^{-1} R^{-1} Q^{-1} \bar{M}^{-2} (B_{\bar{\boldsymbol{\theta}}} + \eta)^{-4\bar{L}}\}\,,$$

where T is the total number of iterations. Then with probability at least

$$1 - \bar{\mathfrak{m}} R \Big[1 - \delta^{\bar{W}(\bar{W}+1)\bar{L}} (2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^Q\,,$$

the optimization error $\mathcal{E}_{opt}^-$ in Equation 15 satisfies

$$\mathcal{E}_{opt}^- \leq \frac{\bar{M}^2}{2R} + \frac{C_5 \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{3\bar{L}} \eta}{\sqrt{R}} + \frac{C_6 \mathfrak{m} \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{T} + C_7 \bar{M}^2 B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}} \delta\,,$$

where η is the projection radius defined in Equation 10.

3.4 Statistical Error

In this section, we present the upper bound of the statistical error, with detailed proofs provided in Appendix B. Note that

$$\mathcal{E}_{sta} := \sup_{u\in\mathcal{PNN}} \left|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)\right|$$

is a random variable, since it is a function of the Monte Carlo sample points $\{X_p\}_{p=1}^{N_{\text{in}}}$, $\{Y_q\}_{q=1}^{N_b}$. Our task is to control $\mathcal{E}_{sta}$ with high probability.

Theorem 5 *Let $\mathcal{PNN} = \mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$. Let $N_{\text{in}} = N_b = N_s$ in the Monte Carlo sampling. Let $0 < \xi < 1$. Then, with probability at least $1-\xi$, it holds that*

$$\begin{aligned}\mathcal{E}_{sta} &= \sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}} \left|\mathcal{L}(u_{\mathfrak{m},\boldsymbol{\theta}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})\right| \\ &\le C_8 M^2 B_{\boldsymbol{\theta}}^{2L} N_s^{-1/2}\big(\sqrt{\log(B_{\boldsymbol{\theta}} WLN_s)} + \sqrt{\log \xi^{-1}}\big)\,,\end{aligned}$$

where C_8 is a universal constant which only depends on Ω, W, L, d and B_0.

Theorem 5 provides a general analysis for any $\mathcal{PNN}$ class. From Theorem 4, we know that the PGD output $u_{\mathcal{A}}$ belongs to the specific class $\mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$ due to the projection steps that explicitly limit parameter variation. This classification, when combined with Theorem 5, yields precise upper bounds for the statistical error $\mathcal{E}_{sta}$. Critically, the bound established in Theorem 5 remains independent of the number of sub-networks $\mathfrak{m}$, enabling effective control of statistical error even in highly over-parameterized settings where $\mathfrak{m}$ becomes arbitrarily large.

3.5 Proof of Main Result

We now present the proof of Theorem 1, which provides the first comprehensive error analysis integrating approximation, statistical, and optimization errors for PDE solving using over-parameterized deep networks to the best of our knowledge.

Step 1: By Theorem 3, we know that for any $0 < \epsilon \ll 1$, there exists a neural network $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$, with

$$\begin{aligned}\bar{\mathfrak{m}} &= \lceil C_1 \epsilon^{-\frac{d}{n-\mu-1}} \rceil\,, \quad \bar{M} = C_2 \epsilon^{-\frac{3d}{2(n-\mu-1)}}\,, \quad \bar{W} = 2^{\lceil \log_2(d+n-1)\rceil + 1}\,, \\ \bar{L} &= \lceil \log_2(d+n-1)\rceil + 2\,, \quad B_{\bar{\boldsymbol{\theta}}} = C_3 \epsilon^{-\frac{2d+2n}{n-\mu-1}}\,,\end{aligned}$$

such that the approximation error $\mathcal{E}_{app} \le \epsilon^2 < \epsilon$.

Step 2: By Theorem 5, with probability at least $1-\xi$, the statistical error satisfies

$$\mathcal{E}_{sta} \le C_8 \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{2\bar{L}} N_s^{-1/2} \big\{ \log^{1/2}\left[(B_{\bar{\boldsymbol{\theta}}} + \eta)\bar{W}\bar{L}N_s\right] + \log^{1/2}(\xi^{-1}) \big\}\,.$$

Setting

$$N_s = \lceil C \epsilon^{-\tilde{C}_3(\mu,d,\beta_0,n)} \rceil\,, \quad \xi = C\epsilon^{\tilde{C}_3(\mu,d,\beta_0,n)}$$

with

$$\beta_0 = \max\{\beta,\, (2d+2n)(n-\mu-1)^{-1}\}\,, \quad \tilde{C}_3 = 4\beta_0 \log(d+n-1) + \frac{6d}{n-\mu-1} + 12\beta_0 + 2\,,$$

it follows that, with probability at least $1-\xi$, $\mathcal{E}_{sta} \leq C\epsilon \log^{1/2}(C\epsilon^{-1}) = \tilde{\mathcal{O}}(\epsilon)$.

Step 3: Recall that $\eta = \epsilon^{-\beta}, \beta > 0$. In order to bound the optimization error $\mathcal{E}_{opt}^{-} \leq C\epsilon$ with probability at least $1-\xi$, we need to determine parameters Q, R, δ, T, η such that

$$1 - \bar{\mathfrak{m}} R\Big[1 - \delta^{\bar{W}(\bar{W}+1)\bar{L}} (2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^{Q} \geq 1 - \xi\,, \tag{23}$$

$$\frac{\bar{M}^2}{2R} + \frac{C_5 \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{3\bar{L}} \eta}{\sqrt{R}} + \frac{C_6 \mathfrak{m} \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{T} + C_7 \bar{M}^2 B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}} \delta \leq C\epsilon\,. \tag{24}$$

To ensure the first, second and the last terms in Equation 24 bounded by $C\epsilon$, we require

$$\delta \leq C\epsilon^{\frac{6(d+n)\log(d+n-1)+21d+18n}{n-\mu-1}+1}\,, \quad R \geq \lceil C\epsilon^{-6\log(d+n-1)\beta_0 - 18\beta_0 - \frac{6d}{n-\mu-1} - 2\beta - 2} \rceil\,.$$

Since $1 - x \leq x^{-2}$ for $x \geq 0$, we could solve for Q in Equation 23 as follows:

$$Q \geq \sqrt{\frac{\bar{\mathfrak{m}} R}{\xi}} \left(\frac{\delta}{2B_{\bar{\boldsymbol{\theta}}}}\right)^{-\bar{W}(\bar{W}+1)\bar{L}} \Rightarrow Q \geq \epsilon^{-\frac{C(d+n)^3 \log_2^2(d+n-1)}{n-\mu-1} - 5\beta_0 \log_2(d+n-1) - 15\beta_0 - \beta}\,.$$

Then, $\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q \geq \lceil C \cdot \epsilon^{-\tilde{C}_1(\mu,d,\beta,\beta_0,n)} \rceil$. Finally, we turn to the third term of Equation 24. To ensure that

$$\frac{C_6 \mathfrak{m} \bar{M}^2 (B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{T} \leq C\epsilon\,,$$

we require that

$$T \geq \lceil C\epsilon^{-\tilde{C}_2(\mu,d,\beta,\beta_0,n)} \rceil\,.$$

Thus, we complete the proof of our main theorem. □

4. Related Work

This section surveys recent advances in error analysis for deep neural networks. We begin by discussing some seminal contributions of approximation and statistical error analysis, then explore recent developments in optimization error analysis and the theoretical attempts to combine all three types errors, providing the context necessary to understand the motivations and innovations presented in this paper.

4.1 Approximation Error

Approximation error in the context of deep neural networks quantifies the discrepancy between the target function and its neural network representation. Theoretical analysis of approximation capabilities began with shallow sigmoidal networks in the 1980s (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991), as reviewed in Pinkus (1999). Recent years

have witnessed a shift toward ReLU networks due to their superior empirical performance in contemporary learning tasks. Yarotsky (Yarotsky, 2017) pioneered the construction of ReLU networks that achieve arbitrary approximation accuracy using Taylor expansion. This breakthrough inspired modern approximation techniques that control errors through architectural parameters such as depth, width, and network size (Yarotsky, 2017, 2018; Petersen and Voigtlaender, 2018; Zhou, 2020; Shen et al., 2020; Siegel and Xu, 2020; Shen et al., 2022; Lu et al., 2021a). For comprehensive reviews, see Petersen (2020); DeVore et al. (2021).

Researchers have also developed neural networks with exceptional expressivity that mitigate the curse of dimensionality (Yarotsky and Zhevnerchuk, 2020; Yarotsky, 2021; Shen et al., 2021b,a; Jiao et al., 2023b). These approximation theories have recently extended to Sobolev spaces, particularly for deep learning solutions to PDEs. Gühring et al. adapted Yarotsky's approach to Sobolev spaces using approximate partition of unity and averaged Taylor expansion (Gühring et al., 2020; Gühring and Raslan, 2021). Additionally, researchers have derived Sobolev norm approximations for networks with ReLU^k activations by examining connections between deep neural networks and B-splines (Duan et al., 2022; Jiao et al., 2023c). Recent work has further established links between deep neural network approximation capabilities and Barron spaces, yielding results that overcome the curse of dimensionality (Ma et al., 2022; Liao and Ming, 2023).

4.2 Statistical Error

Statistical (generalization) error in learning theory is characterized through the uniform law of large numbers over the network class. Classical approaches in empirical process theory use tools such as symmetrization and Lipschitz contraction to transform generalization error analysis into bounding the complexity of neural network classes, measured by Rademacher complexity, covering number, or VC-dimension. For comprehensive treatments, see Van Der Vaart and Wellner (1996); van de Geer (2000); Giné and Nickl (2021); Cucker and Smale (2002). However, generalization analysis based solely on the uniform law of large numbers may yield suboptimal error bounds (Bartlett et al., 2017). Localized techniques that leverage the local structure of hypothesis function classes can achieve sharper error bounds when the Bernstein condition or offset condition is satisfied, as demonstrated in Bartlett et al. (2005); Koltchinskii (2006); Mendelson (2018); Xu and Zeevi (2021); Kanade et al. (2024) and related literature.

It is important to note that the statistical error in the deep Ritz method (DRM) cannot be directly addressed using the contraction principle, as the differential operators in the loss function lack Lipschitz continuity. One effective approach to overcome this challenge involves using the chain rule to represent the gradient of the neural network as another neural network class, then bounding the complexity of this derived class (Duan et al., 2022). By recasting the gradient representation in this manner, researchers can exploit properties of neural network classes—such as Lipschitz continuity and covering numbers—to establish bounds on statistical error. This methodology provides a rigorous framework for analyzing the generalization performance of deep PDE solvers (Jiao et al., 2022; Duan et al., 2022; Lu et al., 2021c; Jiao et al., 2023f; Ji et al., 2024; Yang and He, 2024; Jiao et al., 2023c).

4.3 Theory on ERM with Deep Neural Networks

The convergence rate of Empirical Risk Minimization (ERM) with deep neural networks in regression, classification, and PDE solutions can be established within nonparametric estimation frameworks (Bauer and Kohler, 2019; Kohler and Langer, 2021; Schmidt-Hieber et al., 2020; Nakada and Imaizumi, 2020; Farrell et al., 2021; Jiao et al., 2023d; Suzuki, 2018; Suzuki and Nitanda, 2021; Weinan et al., 2019; Hong et al., 2021; Lu et al., 2021d; Hutzenthaler et al., 2020; Shin, 2020; Lanthaler et al., 2022; Müller and Zeinhofer, 2021; Mishra and Rusch, 2021; Kutyniok et al., 2022; Son et al., 2021; Wang et al., 2022; Weinan et al., 2020; Jiao et al., 2022; Duan et al., 2022; Lu et al., 2021c; Mishra and Molinaro, 2022; Ji et al., 2024; Yang and He, 2024; Hu et al., 2023, 2024; Dai et al., 2023; Yu et al., 2024). This approach carefully balances the trade-off between approximation and statistical errors, providing theoretical guarantees on deep learning model performance. However, these convergence rate results for ERM are only applicable when the neural network class size is smaller than the training sample count. For deep ReLU networks, their VC-dimension or covering number is bounded by their size (Bartlett et al., 2019), meaning theoretical convergence guarantees for ERM can only be established in under-parameterized regimes, where depth, width, and network size are selected as functions of sample size to balance approximation and statistical errors.

Recent work by Jiao et al. (2023e); Yang and Zhou (2024); Chen et al. (2025); Jiao et al. (2024); Ding et al. (2025) proposes using weight norm instead of conventional measures like network width, depth, or size to characterize both approximation and statistical errors. This approach has enabled researchers to derive convergence rates for ERM in over-parameterized regimes, representing a significant advancement toward theoretically understanding modern, over-parameterized neural networks without addressing optimization error.

4.4 Optimization Error

Current analytical frameworks for studying optimization error in deep neural networks primarily rely on neural tangent kernel and mean field theory in over-parameterized or infinite width settings (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou and Gu, 2019; Liu et al., 2022; Chizat et al., 2019; Nguyen, 2021; Lu et al., 2020; Mahankali et al., 2024). A significant limitation of these approaches is their requirement that training dynamics remain proximate to initial parameter values, a condition rarely satisfied in practical applications.

In over-parameterized regimes, the optimal training loss approaches zero as the network perfectly interpolates the data. This phenomenon presents a dual challenge: on one hand, when networks memorize data, they may develop substantially large weight upper bounds (Vershynin, 2020; Vardi et al., 2021; Shen et al., 2020; Lu et al., 2021a), potentially rendering size-independent generalization error uncontrollable (Golowich et al., 2018; Jiao et al., 2023e). On the other hand, the norm constraints employed in recent ERM analyses (Jiao et al., 2023e; Yang and Zhou, 2024; Chen et al., 2025; Jiao et al., 2024) may be insufficiently large, and even randomized initializations in NTK and mean field frameworks may violate these constraints. These challenges represent the primary obstacles preventing researchers from simultaneously addressing the three fundamental errors—approximation, generalization, and optimization—in modern over-parameterized deep learning.

4.5 Works on Complete Error Analysis

Beck et al. (2022); Jentzen and Welti (2023) conducted a comprehensive error analysis of deep regression in the under-parameterization setting. Building upon recent research on the estimation error of gradient descent in regression (Kohler and Langer, 2021; Kohler and Krzyzak, 2023b; Drews and Kohler, 2023), Jiao et al. (2025) derived consistency results for DRM, encompassing all three types of errors. However, it should be noted that the results in Jiao et al. (2025) are specifically applicable to three-layer networks only. One major drawback of Jiao et al. (2025); Kohler and Langer (2021); Kohler and Krzyzak (2023b); Drews and Kohler (2023) is their reliance on the iteration being very close to the initialization, which is an unrealistic requirement. This restrictive condition limits the practicality of their proposed theories, as real-world training of deep neural networks often involves substantial parameter updates that move far from the initial configuration.

In contrast, based on the techniques proposed in Kohler and Krzyzak (2023a), we obtain consistency results for DRM that simultaneously addresses all three types of error in the over-parameterized deep neural network setting while allowing parameter iterates to diverge from their initialization values. This relaxation substantially enhances the practical relevance of our theoretical framework.

5. Conclusion

In this paper, we provide the first complete error analysis for DRM that includes the approximation, statistical, and optimization error in the scenario of over-parameterization. Our analysis is based on the projected gradient descent algorithm and does not require constraining the neural network weights near their initial values during the optimization process, thereby completely moving away from the lazy training framework. This marks a milestone in the field of theoretical understanding of solving PDEs via deep learning.

Several questions deserve further investigation. Firstly, our analytical techniques rely on the random initialization of over-parameterized neural networks. In the current analysis, we do not use any prior knowledge to design the parameter initialization method; instead, we use a general uniform distribution. This results in a theoretically excessive number of training samples and iteration steps to achieve the desired accuracy. Therefore, exploring the effective use of prior information to improve these results is an intriguing subject. Secondly, the gradient descent algorithm used in our theoretical analysis is full gradient descent, which has certain gaps compared to the stochastic gradient descent (SGD) algorithm commonly used in practice. Lastly, recent second-order methods, such as Müller and Zeinhofer (2023); Müller and Montúfar (2024), have demonstrated high accuracy in deep PDE solving, and analyzing these methods within our framework presents a promising and challenging direction for future research. Overall, the analytical framework presented in this paper is highly versatile and can be applied to other areas in deep PDE solving, such as PINNs and various inverse problems. We aim to explore these topics in greater depth.

Acknowledgments

This work has been funded by the National Key Research and Development Program of China (No. 2023YFA1000103), by the National Natural Science Foundation of China (No. 123B2019, No. 12125103, No. U24A2002, No. 12371441), and by the Fundamental Research Funds for the Central Universities.

The appendix is divided into four parts. In Appendix A, we provide a detailed explanation of how to construct the optimal approximation function in the Sobolev space using the $\mathcal{PNN}$ structure as discussed in Section 3.2. In Appendix B, we present the complete proof of the statistical error upper bound estimation for the over-parameterized $\mathcal{PNN}$ network class as given in Section 3.4. Note that some of the lemmas used in Appendices A and B are derived from previous work or are classical results. For the sake of completeness, we have still provided detailed proofs of these lemmas. In Appendix C, we present the proofs of the theorems and lemmas involved in the optimization error analysis in Section 3.3, which are newly proposed in this paper. In Appendix D, we evaluate the complexity of a single step of the projected gradient descent algorithm.

Appendix A. Detailed Approximation Error Analysis

This section presents a detailed supplement to Section 3.2 of the main text. Building on the techniques developed in Yarotsky (2017); Jiao et al. (2023e); Gühring and Raslan (2021), we establish an upper bound for the approximation error of neural networks in the Sobolev norm $W^{1,p}$, assuming the target function belongs to $W^{n,p}$ with $n \geq 2$. This bound is obtained by explicitly constructing tanh-activated neural networks that approximate local Taylor polynomials. A key feature of our construction is its parallel architecture: the final network is formed as a linear combination of multiple structurally similar, fully connected sub-networks, as described in Section 2.2.

A.1 Exponential Partition of Unity with tanh Activation

We first introduce the notion of 'exponential partition of unity', following the construction in Gühring and Raslan (2021). This concept plays a crucial role in localizing function approximations and is particularly compatible with the tanh activation function.

Definition 2 *Let $d, j, \tau, N \in \mathbb{N}$ and $s \in \mathbb{R}$. A set of function families $(\Lambda^{(j,\tau,N,s)})_{N\in\mathbb{N},s\in\mathbb{R}_{\geq 1}}$, where each*

$$\Lambda^{(j,\tau,N,s)} := \{\Psi_m^s : \boldsymbol{m} \in \{0,\dots,N\}^d\}$$

consists of $(N+1)^d$ functions $\Psi_{\boldsymbol{m}}^s : \mathbb{R}^d \to \mathbb{R}$, is called an exponential partition of unity of order τ and smoothness j, if there exist constants $D > 0$, $S > 0$, and $C = C(k,d) > 0$, such that for all $N \in \mathbb{N}$, $s \geq S$, and $k \in \{0,\dots,j\}$, the following properties are satisfied:

i. Uniform sobolev bounds:

$$\|\Psi_{\boldsymbol{m}}^s\|_{W^{k,\infty}(\mathbb{R}^d)} \leq CN^k s^{\max\{0,k-\tau\}} \quad \text{for every } \Psi_{\boldsymbol{m}}^s \in \Lambda^{(j,\tau,N,s)}.$$

ii. Exponential decay outside support: Let

$$\Omega_{\boldsymbol{m}}^c = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - N^{-1}\boldsymbol{m}\|_\infty \geq N^{-1}\}.$$

Then for every $\Psi_{\boldsymbol{m}}^s \in \Lambda^{(j,\tau,N,s)}$,

$$\|\Psi_{\boldsymbol{m}}^s\|_{W^{k,\infty}(\Omega_{\boldsymbol{m}}^c)} \leq CN^k s^{\max\{0,k-\tau\}} e^{-Ds}\,.$$

iii. Approximate partition of unity:

$$\left\| \mathbf{1}_{(0,1)^d} - \sum_{\boldsymbol{m}\in\{0,\dots,N\}^d} \Psi_{\boldsymbol{m}}^s \right\|_{W^{k,\infty}((0,1)^d)} \leq CN^k s^{\max\{0,k-\tau\}} e^{-Ds}\,,$$

for every $\Psi_{\boldsymbol{m}}^s \in \Lambda^{(j,\tau,N,s)}$.

iv. Neural network realizability: There exists an activation function $\rho : \mathbb{R} \to \mathbb{R}$ *such that for each* $\Psi_{\boldsymbol{m}}^s \in \Lambda$, *there exists a neural network* $\psi_{\boldsymbol{\theta}}$ *with* d*-dimensional input and output, two hidden layers, and at most* C *nonzero weights, satisfying*

$$\prod_{l=1}^{d} [\psi_{\boldsymbol{\theta}}(\boldsymbol{x})]_l = \Psi_{\boldsymbol{m}}^s, \quad \text{and} \quad \|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})\|_{W^{k,\infty}((0,1)^d)} \leq CN^k \cdot s^{\max\{0,k-\tau\}}\,.$$

Moreover, the network weights satisfy $\|\boldsymbol{\theta}\|_\infty \leq CsN$.

Let $\rho = \tanh$. We now define a class of smooth bump functions, denoted by $\Lambda^{(j,0,N,s)}(\rho)$.

Definition 3 *Let* $j \in \mathbb{N}$, $\tau = 0$, *and let* $\rho = \tanh$, *defined by* $\rho(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. *For a scaling factor* $s \geq 1$, *we define the one-dimensional bump function* $\psi^s : \mathbb{R} \to \mathbb{R}$ *as*

$$\psi^s(x) := \frac{\rho(s(x+3/2)) - \rho(s(x-3/2))}{2}\,.$$

Given $N, d \in \mathbb{N}$ *and* $\boldsymbol{m} \in \{0,\dots,N\}^d$, *we construct the* d*-dimensional bump function* $\Psi_{\boldsymbol{m}}^s : \mathbb{R}^d \to \mathbb{R}$ *as a tensor product of scaled and shifted versions of* ψ^s*:*

$$\Psi_{\boldsymbol{m}}^s(\boldsymbol{x}) := \prod_{l=1}^{d} \psi^s\Big(3N\Big(x_l - \frac{m_l}{N}\Big)\Big)\,. \tag{25}$$

Finally, for $s \geq 1$, *we denote the collection of such bump functions by* $\Lambda^{(j,0,N,s)}(\rho) := \{\Psi_{\boldsymbol{m}}^s : \boldsymbol{m} \in \{0,\dots,N\}^d\}$.

The following result, as shown in Gühring and Raslan (2021) (Lemma 4.5), confirms that $\Lambda^{(j,0,N,s)}(\rho)$ satisfies all the conditions to form an exponential partition of unity of order 0 and smoothness j.

Lemma 1 *The collection of families of functions* $(\Lambda^{(j,0,N,s)}(\rho))_{N\in\mathbb{N},s\in\mathbb{R}_{\geq 1}}$ *defined in Definition 3 is an exponential partition of unity of order* 0 *and smoothness* j.

A.2 Approximate the Target with Polynomials

To construct a local approximation of Sobolev functions using a smooth partition of unity and polynomials, we present the following result, which serves as the foundation for the neural network approximation developed in subsequent sections. This proposition is adapted from Gühring and Raslan (2021) (Lemma D.1), where a detailed proof is provided.

Proposition 4 *Let $d \in \mathbb{N}$, $j \in \mathbb{N}$, $k \in \{0, \dots, j\}$, $k \leq n-1$, and $1 \leq p \leq \infty$. Let $\mu > 0$ be arbitrarily small, and let $\rho(x) = \tanh(x)$. For $\boldsymbol{\alpha} \in \mathbb{N}^d$, define $\boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdots x_d^{\alpha_d}$. For $N \in \mathbb{N}$, set $s = N^\mu$. Let $(\Psi_{\boldsymbol{m}}^s)_{\boldsymbol{m} \in \{0,\dots,N\}^d}$ be a family of partition functions in $\Lambda^{(j,0,N,s)}(\rho)$. Then there exist constants $C = C(d, n, p, k) > 0$ and $\widetilde{N} = \widetilde{N}(d, p, \mu, k) \in \mathbb{N}$ such that the following holds: For every $f \in W^{n,p}(\Omega)$, there exists a function f_N defined as*

$$f_N := \sum_{\boldsymbol{m} \in \{0,\dots,N\}^d} \sum_{\|\boldsymbol{\alpha}\|_1 \leq n-1} c_{f,\boldsymbol{m},\boldsymbol{\alpha}} \Psi_{\boldsymbol{m}}^s \boldsymbol{x}^{\boldsymbol{\alpha}} \tag{26}$$

satisfies the approximation estimate

$$\|f - f_N\|_{W^{k,p}(\Omega)} \leq C \|f\|_{W^{n,p}(\Omega)} \cdot N^{-(n-k-\mu k)}$$

for all $N \geq \widetilde{N}$. In addition, the polynomial coefficients satisfy

$$|c_{f,\boldsymbol{m},\boldsymbol{\alpha}}| \leq C \|\tilde{f}\|_{W^{n,p}(\Omega_{\boldsymbol{m},N})} N^{d/p},$$

where $\Omega_{\boldsymbol{m},N} \in \mathbb{R}^d$ is the open cube in terms of $\|\cdot\|_\infty$, centered at $N^{-1}\boldsymbol{m}$ with radius N^{-1}, and $\tilde{f} \in W^{n,p}(\mathbb{R}^d)$ is an extension of f.

Remark 9 *There exists some $C = C(n, d, p) > 0$ such that $\|\tilde{f}\|_{W^{n,p}(\mathbb{R}^d)} \leq C \|f\|_{W^{n,p}(\Omega)}$ (see Stein (1970), Theorem VI.3.1.5).*

A.3 Approximate Polynomials with Neural Networks

This subsection aims to illustrate how neural networks can be used to approximate sums of localized polynomials. We will specifically analyze this approximation capability of neural networks under the $W^{1,p}$ norm (i.e., the case $k = 1$), based on the definition of approximation error in Equation 14. It is worth noting that for $k \geq 2$, corresponding approximation results can also be obtained using similar techniques.

The key to achieving this approximation lies in the adopted activation function $\rho(x) = \tanh(x)$ and its properties. A core characteristic of this function is its smoothness and the existence of non-zero derivatives: there exists a point $x_0 \in \mathbb{R}$ and a neighborhood $\mathcal{U}$ such that $\rho \in C^{i+1}(\mathcal{U})$ and $\rho^{(r)}(x_0) \neq 0$ for all $r = 1, \dots, i$. As shown in Gühring and Raslan (2021) (Proposition 4.7), the monomials x^r can be effectively approximated by linear combinations of shifted and scaled copies of ρ. The following lemma formalizes this result in the specific cases of x and x^2.

Lemma 2 *Let $B > 0$, and define $\rho(x) = \tanh(x)$. Suppose $x_0 \in \mathbb{R}$ is a point such that $\rho^{(r)}(x_0) \neq 0$ for $r = 1, 2$. Let $C = C(B) > 0$ be a constant, where $C(B)$ is monotonically increasing in B, and let $\epsilon \in (0, 1)$ denote the desired approximation accuracy. Then,*

1. *(Approximation of x) There exists a neural network $\phi_{\boldsymbol{\theta}_1} \in \mathcal{NN}(1, 2, C'\epsilon^{-1})$ with parameters $\boldsymbol{\theta}_1 = ((\boldsymbol{A}_0, \boldsymbol{b}_0), (\boldsymbol{A}_1, \boldsymbol{b}_1))$ defined as*

$$\boldsymbol{A}_0 = \Big(-\frac{\epsilon}{C} \Big), \quad \boldsymbol{b}_0 = (x_0)\,, \quad \boldsymbol{A}_1 = \Big(-\frac{C}{\epsilon \rho^{(1)}(x_0)} \Big), \quad \boldsymbol{b}_1 = \frac{C\rho(x_0)}{\epsilon \rho^{(1)}(x_0)}\,,$$

where $C' = C'(B) > 0$, such that

$$\|x - \phi_{\boldsymbol{\theta}_1}(x)\|_{W^{1,\infty}([-B,B])} \leq \epsilon\,. \tag{27}$$

2. *(Approximation of x^2) There exists a neural network $\phi_{\boldsymbol{\theta}_2} \in \mathcal{NN}(2, 2, C''\epsilon^{-2})$ with parameters $\boldsymbol{\theta}_2 = ((\boldsymbol{A}_0, \boldsymbol{b}_0), (\boldsymbol{A}_1, \boldsymbol{b}_1))$ defined as*

$$\boldsymbol{A}_0 = \begin{pmatrix} -\frac{\epsilon}{C} \\ -\frac{2\epsilon}{C} \end{pmatrix}, \quad \boldsymbol{b}_0 = \begin{pmatrix} x_0 \\ x_0 \end{pmatrix}, \quad \boldsymbol{A}_1 = \frac{C^2}{\epsilon^2 \rho^{(2)}(x_0)} \begin{pmatrix} -2 & 1 \end{pmatrix}, \quad \boldsymbol{b}_1 = \frac{C^2 \rho(x_0)}{\epsilon^2 \rho^{(2)}(x_0)}\,,$$

where $C'' = C''(B) > 0$, such that

$$\|x^2 - \phi_{\boldsymbol{\theta}_2}(x)\|_{W^{1,\infty}([-B,B])} \leq \epsilon\,. \tag{28}$$

To extend the approximation from individual monomials to product functions, we employ the polarization identity $xy = [(x+y)^2 - (x-y)^2]/4$, which transforms the problem into the approximation of squared terms.

Further error analysis requires handling composite and product functions in $W^{1,\infty}$ spaces. Relevant technical arguments are shown in Lemma 3 below:

Lemma 3 *Let $d_1, d_2 \in \mathbb{N}$, and let $\Omega_1 \subset \mathbb{R}^{d_1}$, $\Omega_2 \subset \mathbb{R}^{d_2}$ be open, bounded, and convex domains. Then there exists $C_1 = C(d_1, d_2) > 0$ and $C_2 = C(d_1) > 0$ such that*

1. *(Chain rule) Let $\boldsymbol{f} \in W^{1,\infty}(\Omega_1; \mathbb{R}^{d_2})$ and $g \in W^{1,\infty}(\Omega_2)$ be Lipschitz functions with $\mathrm{range}(\boldsymbol{f}) \subset \Omega_2$. Then $g \circ \boldsymbol{f} \in W^{1,\infty}(\Omega_1)$, and*

$$\|g \circ \boldsymbol{f}\|_{W^{1,\infty}(\Omega_1)} \leq C_1 \max\Big\{ \|g\|_{L^\infty(\Omega_2)},\ |g|_{W^{1,\infty}(\Omega_2)} \cdot |\boldsymbol{f}|_{W^{1,\infty}(\Omega_1;\mathbb{R}^{d_2})} \Big\}. \tag{29}$$

2. *(Product rule) Let $u, v \in W^{1,\infty}(\Omega_1)$. Then $uv \in W^{1,\infty}(\Omega_1)$, and*

$$\|uv\|_{W^{1,\infty}(\Omega_1)} \leq C_2 \|u\|_{W^{1,\infty}(\Omega_1)} \|v\|_{W^{1,\infty}(\Omega_1)}\,. \tag{30}$$

Moreover, if $u \in W^{1,\infty}(\Omega_1)$, $v \in W^{1,\infty}(\Omega_2)$, and $h(\boldsymbol{x}, \boldsymbol{y}) := u(\boldsymbol{x})v(\boldsymbol{y})$ for $(\boldsymbol{x}, \boldsymbol{y}) \in \Omega_1 \times \Omega_2$, then $h \in W^{1,\infty}(\Omega_1 \times \Omega_2)$ and

$$\|h\|_{W^{1,\infty}(\Omega_1 \times \Omega_2)} \leq \max\big\{ \|u\|_{W^{1,\infty}(\Omega_1)} \|v\|_{L^\infty(\Omega_2)},\ \|u\|_{L^\infty(\Omega_1)} \|v\|_{W^{1,\infty}(\Omega_2)} \big\}. \tag{31}$$

Proof The first part is a direct consequence of Gühring et al. (2020) (Corollary B.5). For the general case of the second part, the product rule estimate follows from Gühring and Raslan (2021) (Lemma B.5). For the case of separated variables, it is straightforward

to verify that $\|h\|_{L^\infty(\Omega_1\times\Omega_2)} = \|u\|_{L^\infty(\Omega_1)}\|v\|_{L^\infty(\Omega_2)}$. Moreover, for all $1 \le i \le d_1$ and $1 \le j \le d_2$, we have $\partial_{x_i} h(\boldsymbol{x},\boldsymbol{y}) = \partial_{x_i} u(\boldsymbol{x}) \cdot v(\boldsymbol{y})$ and $\partial_{y_j} h(\boldsymbol{x},\boldsymbol{y}) = u(\boldsymbol{x}) \cdot \partial_{y_j} v(\boldsymbol{y})$. Therefore,

$$\begin{aligned}\|h\|_{W^{1,\infty}(\Omega_1\times\Omega_2)} &= \max\big\{\|h\|_{L^\infty(\Omega_1\times\Omega_2)}, \max_{1\le i\le d_1}\|\partial_{x_i}h\|_{L^\infty}, \max_{1\le j\le d_2}\|\partial_{y_j}h\|_{L^\infty}\big\}\\ &\le \max\big\{\|u\|_{W^{1,\infty}(\Omega_1)}\|v\|_{L^\infty(\Omega_2)}, \|u\|_{L^\infty(\Omega_1)}\|v\|_{W^{1,\infty}(\Omega_2)}\big\}.\end{aligned}$$

This concludes the proof of the second part. ■

With the above preparations, we now construct neural networks to approximate products of variables. We begin with the basic case of approximating the binary product xy, as described in the following lemma.

Lemma 4 *(Approximation of xy) Let $B > 0$, $C = C(B) > 0$, $C' = C'(B) > 0$ and $\epsilon \in (0,1)$ be the desired approximation accuracy. Then, there exists a neural network $\phi_{\boldsymbol{\theta}} \in \mathcal{NN}(4, 2, C'\epsilon^{-2})$ with parameters*

$$\begin{aligned}\boldsymbol{A}_0 &= \begin{pmatrix} -\frac{\epsilon}{C} & -\frac{2\epsilon}{C} & -\frac{\epsilon}{C} & -\frac{2\epsilon}{C} \\ -\frac{\epsilon}{C} & -\frac{2\epsilon}{C} & \frac{\epsilon}{C} & \frac{2\epsilon}{C}\end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{4\times 2}, \qquad \boldsymbol{b}_0 = (x_0, x_0, x_0, x_0)^{\mathrm{T}} \in \mathbb{R}^4,\\ \boldsymbol{A}_1 &= \frac{C^2}{4\epsilon^2\rho^{(2)}(x_0)}(-2,1,2,-1) \in \mathbb{R}^{1\times 4}, \qquad \boldsymbol{b}_1 = 0 \in \mathbb{R},\end{aligned} \tag{32}$$

such that

$$\|xy - \phi_{\boldsymbol{\theta}}(x,y)\|_{W^{1,\infty}([-B,B]^2)} \le \epsilon.$$

Proof To approximate xy, we use the polarization identity. Let $f : [-2B, 2B] \to \mathbb{R}$ be the square function, $f(z) = z^2$. Define auxiliary linear functions:

$$\begin{aligned}u : [-B,B]^2 \to [-2B,2B],\ (x,y) \mapsto x + y,\\ v : [-B,B]^2 \to [-2B,2B],\ (x,y) \mapsto x - y.\end{aligned}$$

Then, we have $xy = (f \circ u(x,y) - f \circ v(x,y))/4$ for all $(x,y) \in [-B,B]^2$. By Lemma 2, let $\phi_{\hat{\boldsymbol{\theta}}}(z)$ be the neural network satisfying

$$\|f - \phi_{\hat{\boldsymbol{\theta}}}\|_{W^{1,\infty}([-2B,2B])} \le \tilde{\epsilon}.$$

The explicit form of $\phi_{\hat{\boldsymbol{\theta}}}(z)$ is given by

$$\phi_{\hat{\boldsymbol{\theta}}}(z) = \frac{\tilde{C}^2}{\tilde{\epsilon}^2\rho^{(2)}(x_0)}\Big[\rho(x_0) - 2\rho\Big(x_0 - \frac{\tilde{\epsilon}}{\tilde{C}}z\Big) + \rho\Big(x_0 - \frac{2\tilde{\epsilon}}{\tilde{C}}z\Big)\Big],$$

where $\tilde{C} = C_{\mathrm{Lem}\,2}(2B)$ and x_0 is a suitable point where $\rho^{(2)}(x_0) \neq 0$. We then construct the neural network $\phi_{\boldsymbol{\theta}}(x,y)$ to approximate xy using this $\phi_{\hat{\boldsymbol{\theta}}}$ and the polarization identity:

$$\phi_{\boldsymbol{\theta}}(x,y) = \frac{1}{4}\big(\phi_{\hat{\boldsymbol{\theta}}}(x+y) - \phi_{\hat{\boldsymbol{\theta}}}(x-y)\big) = \frac{1}{4}\big(\phi_{\hat{\boldsymbol{\theta}}} \circ u(x,y) - \phi_{\hat{\boldsymbol{\theta}}} \circ v(x,y)\big).$$

The approximation error for xy on $[-B,B]^2$ is bounded as follows:

$$\begin{aligned}\big\|\phi_{\boldsymbol{\theta}}(x,y)-xy\big\|_{W^{1,\infty}([-B,B]^2)} &= \frac{1}{4}\big\|\phi_{\hat{\boldsymbol{\theta}}}\circ u-\phi_{\hat{\boldsymbol{\theta}}}\circ v-(f\circ u-f\circ v)\big\|_{W^{1,\infty}([-B,B]^2)}\\ &\le \frac{1}{4}\big\|\phi_{\hat{\boldsymbol{\theta}}}\circ u-f\circ u\big\|_{W^{1,\infty}([-B,B]^2)}+\frac{1}{4}\big\|\phi_{\hat{\boldsymbol{\theta}}}\circ v-f\circ v\big\|_{W^{1,\infty}([-B,B]^2)}.\end{aligned}$$

Note that $|u|_{W^{1,\infty}([-B,B]^2)}=|v|_{W^{1,\infty}([-B,B]^2)}=1$. Applying the chain rule estimate Equation 29 from Lemma 3 to each term, we have

$$\|\phi_{\boldsymbol{\theta}}(x,y)-xy\|_{W^{1,\infty}([-B,B]^2)}\le C_{\text{chain}}\tilde{\epsilon}\,.$$

Setting $\tilde{\epsilon}=\epsilon/C_{\text{chain}}$ then yields $\|\phi_{\boldsymbol{\theta}}(x,y)-xy\|_{W^{1,\infty}([-B,B]^2)}\le\epsilon$. Letting $C:=C_{\text{chain}}\tilde{C}$, we obtain the explicit expression

$$\begin{aligned}\phi_{\boldsymbol{\theta}}(x,y)=\frac{C^2}{4\epsilon^2\rho^{(2)}(x_0)}\Big[&-2\rho\Big(x_0-\frac{\epsilon}{C}(x+y)\Big)+\rho\Big(x_0-\frac{2\epsilon}{C}(x+y)\Big)\\ &+2\rho\Big(x_0-\frac{\epsilon}{C}(x-y)\Big)-\rho\Big(x_0-\frac{2\epsilon}{C}(x-y)\Big)\Big].\end{aligned}$$

The parameters defined in Equation 32 directly implement this construction. ■

Building on the approximation of the bivariate monomial xy, we next construct approximations for the multivariate product $x_1\cdots x_d$.

Lemma 5 *(Approximation of $x_1\cdots x_d$) Let $d\ge 2$, $0<\epsilon<1$, and set $\kappa=\lceil\log_2 d\rceil$. Let $C=C(d)>0$. Then there exists a neural network $\phi_{\boldsymbol{\theta}}\in\mathcal{NN}(2^{\kappa+1},\kappa+1,C\epsilon^{-2})$ such that*

$$\|x_1\cdots x_d-\phi_{\boldsymbol{\theta}}(\boldsymbol{x})\|_{W^{1,\infty}([0,1]^d)}\le 4^d\epsilon\,,\quad \boldsymbol{x}=(x_1,\cdots,x_d)\in[0,1]^d\,.$$

Proof The proof proceeds by building approximations for products of increasing dimensions, beginning with the bivariate case and extending through recursive composition.

We first establish the case where $d=2^\kappa$ for some $\kappa\in\mathbb{N}$. Let $E_s\epsilon$ denote the upper bound of the approximation error at recursion level s for $1\le s\le\kappa$, with $E_0=0$. In our recursive construction, the outputs of one approximation stage $\phi^s_{\boldsymbol{\theta}}$ become the inputs for the next stage. As will be detailed in Step 2, the L^∞ norm of $\phi^s_{\boldsymbol{\theta}}(x_1,\ldots,x_{2^s})$ for $x_i\in[0,1]$ is bounded by $1+E_s\epsilon$. To guarantee that these intermediate outputs remain within a range $[-B,B]$, we need to set $B\ge\max_{0\le s\le\kappa-1}(1+E_s\epsilon)$.

Step 1: Base case ($d=2$). By Lemma 4, there exists $C=C(B)>0$, $C'=C'(B)>0$, and $\phi^1_{\boldsymbol{\theta}}\in\mathcal{NN}(4,2,C'\epsilon^{-2})$ such that

$$\|xy-\phi^1_{\boldsymbol{\theta}}(x,y)\|_{W^{1,\infty}([-B,B]^2)}\le\epsilon\,. \tag{33}$$

Using Equation 33, since $B\ge 1$, we directly have

$$\|x_1x_2-\phi^1_{\boldsymbol{\theta}}(x_1,x_2)\|_{W^{1,\infty}([0,1]^2)}\le E_1\epsilon,\quad E_1=1 \tag{34}$$

Step 2: Recursive step for $d=4$. For $s=2$, denote by

$$\phi^2_{\boldsymbol{\theta}}(x_1,x_2,x_3,x_4):=\phi^1_{\boldsymbol{\theta}}\big(\phi^1_{\boldsymbol{\theta}}(x_1,x_2),\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\big)\,.$$

The approximation error decomposes as

$$\begin{aligned}\|x_1x_2x_3x_4-\phi^2_{\boldsymbol{\theta}}(x_1,x_2,x_3,x_4)\|_{W^{1,\infty}([0,1]^4)} \le & \underbrace{\|x_1x_2x_3x_4-x_1x_2\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\|_{W^{1,\infty}([0,1]^4)}}_{H_1}\\ &+\underbrace{\|x_1x_2\phi^1_{\boldsymbol{\theta}}(x_3,x_4)-\phi^1_{\boldsymbol{\theta}}(x_1,x_2)\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\|_{W^{1,\infty}([0,1]^4)}}_{H_2}\\ &+\underbrace{\|\phi^1_{\boldsymbol{\theta}}(x_1,x_2)\phi^1_{\boldsymbol{\theta}}(x_3,x_4)-\phi^1_{\boldsymbol{\theta}}\big(\phi^1_{\boldsymbol{\theta}}(x_1,x_2),\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\big)\|_{W^{1,\infty}([0,1]^4)}}_{H_3}.\end{aligned}$$

First, by Equations 33 and 34, we have the following bounds

$$\begin{aligned}&\|u-\partial_v\phi^1_{\boldsymbol{\theta}}(u,v)\|_{L^\infty([-B,B]^2)}\le E_1\epsilon\,,\qquad \|v-\partial_u\phi^1_{\boldsymbol{\theta}}(u,v)\|_{L^\infty([-B,B]^2)}\le E_1\epsilon\,,\\ &\|\phi^1_{\boldsymbol{\theta}}\|_{L^\infty([0,1]^2)}\le 1+E_1\epsilon\,,\quad \|\partial_u\phi^1_{\boldsymbol{\theta}}\|_{L^\infty([0,1]^2)}\le 1+E_1\epsilon\,,\quad \|\partial_v\phi^1_{\boldsymbol{\theta}}\|_{L^\infty([0,1]^2)}\le 1+E_1\epsilon\,.\end{aligned}$$

Then, the three error terms are bounded as follows

- H_1: Let $u_1=x_1x_2$, $v_1=x_3x_4-\phi^1_{\boldsymbol{\theta}}(x_3,x_4)$. Then, we have $\|u_1\|_{W^{1,\infty}([0,1]^2)}\le 1$ and $\|v_1\|_{W^{1,\infty}([0,1]^2)}\le E_1\epsilon$. By Equation 31, it holds that

$$H_1\le \max\big\{\|u_1\|_{W^{1,\infty}([0,1]^2)}\|v_1\|_{L^\infty([0,1]^2)},\ \|u_1\|_{L^\infty([0,1]^2)}\|v_1\|_{W^{1,\infty}([0,1]^2)}\big\}\le E_1\epsilon\,.$$

- H_2: Let $u_2=x_1x_2-\phi^1_{\boldsymbol{\theta}}(x_1,x_2)$, $v_2=\phi^1_{\boldsymbol{\theta}}(x_3,x_4)$. Then, $\|u_2\|_{W^{1,\infty}([0,1]^2)}\le E_1\epsilon$, $\|v_2\|_{W^{1,\infty}([0,1]^2)}\le 1+E_1\epsilon$. By Equation 31, we obtain

$$H_2\le E_1\epsilon(1+E_1\epsilon)\,.$$

- H_3: Let $u_3=\phi^1_{\boldsymbol{\theta}}(x_1,x_2)$ and $v_3=\phi^1_{\boldsymbol{\theta}}(x_3,x_4)$. Since $\|u_3\|_{L^\infty([0,1]^2)}\le 1+E_1\epsilon\le B$, it holds that $\|u_3v_3-\phi^1_{\boldsymbol{\theta}}(u_3,v_3)\|_{L^\infty([0,1]^4)}\le\epsilon$. For the first-order partial derivatives, take ∂_{x_1} as an example, we have

$$\begin{aligned}&\big\|\partial_1\phi^1_{\boldsymbol{\theta}}\big(\phi^1_{\boldsymbol{\theta}}(x_1,x_2),\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\big)\partial_{x_1}\phi^1_{\boldsymbol{\theta}}(x_1,x_2)-\partial_{x_1}\phi^1_{\boldsymbol{\theta}}(x_1,x_2)\phi^1_{\boldsymbol{\theta}}(x_3,x_4)\big\|_{L^\infty([0,1]^4)}\\ &\quad\le\big\|\partial_1\phi^1_{\boldsymbol{\theta}}(u_3,v_3)-v_3\big\|_{L^\infty([0,1]^4)}\cdot(1+E_1\epsilon)\le\epsilon(1+E_1\epsilon)\,,\end{aligned}$$

  where ∂_1 denotes taking derivative of the first position. Therefore, we get

$$H_3\le\max\{\epsilon,\epsilon(1+E_1\epsilon)\}\le\epsilon(1+E_1\epsilon)\,.$$

Finally, noting that $E_1=1$ and $0<\epsilon^2<\epsilon<1$, we have

$$\begin{aligned}&\|x_1x_2x_3x_4-\phi^2_{\boldsymbol{\theta}}(x_1,x_2,x_3,x_4)\|_{W^{1,\infty}([0,1]^4)}\\ &\quad\le E_1\epsilon+E_1\epsilon(1+E_1\epsilon)+\epsilon(1+E_1\epsilon)\le[(E_1+1)^2+E_1]\epsilon=E_2\epsilon\,,\quad E_2=5\,.\end{aligned}$$

Step 3: Network realization for $d=4$. According to Lemma 4, the product x_1x_2 (or x_3x_4) can be approximated using an activation layer and a linear layer. To approximate $(x_1x_2)(x_3x_4)$, one would typically repeat this process on the outputs of the two sub-products.

However, since the intermediate results x_1x_2 and x_3x_4 are not needed explicitly, we can merge their final linear layers with the initial linear transformation used to approximate their product. This reduces the number of layers: the full product $x_1x_2x_3x_4$ can be approximated using two activation layers, and a final output layer. Combining this with Equation 32, we construct the weights $((\boldsymbol{A}_0, \boldsymbol{b}_0), (\boldsymbol{A}_1, \boldsymbol{b}_1), (\boldsymbol{A}_2, \boldsymbol{b}_2))$ as follows

$$\boldsymbol{A}_0 = \begin{pmatrix} -\frac{\epsilon}{C} & -\frac{\epsilon}{C} & 0 & 0 \\ -\frac{2\epsilon}{C} & -\frac{2\epsilon}{C} & 0 & 0 \\ -\frac{\epsilon}{C} & \frac{\epsilon}{C} & 0 & 0 \\ -\frac{2\epsilon}{C} & \frac{2\epsilon}{C} & 0 & 0 \\ 0 & 0 & -\frac{\epsilon}{C} & -\frac{\epsilon}{C} \\ 0 & 0 & -\frac{2\epsilon}{C} & -\frac{2\epsilon}{C} \\ 0 & 0 & -\frac{\epsilon}{C} & \frac{\epsilon}{C} \\ 0 & 0 & -\frac{2\epsilon}{C} & \frac{2\epsilon}{C} \end{pmatrix} \in \mathbb{R}^{8\times 4}, \qquad \boldsymbol{b}_0 = \begin{pmatrix} x_0 \\ x_0 \\ x_0 \\ x_0 \\ x_0 \\ x_0 \\ x_0 \\ x_0 \end{pmatrix},$$

$$\boldsymbol{A}_1 = \frac{C}{4\epsilon\rho^{(2)}(x_0)} \begin{pmatrix} 2 & -1 & -2 & 1 & 2 & -1 & -2 & 1 \\ 4 & -2 & -4 & 2 & 4 & -2 & -4 & 2 \\ 2 & -1 & -2 & 1 & -2 & 1 & 2 & -1 \\ 4 & -2 & -4 & 2 & -4 & 2 & 4 & -2 \end{pmatrix} \in \mathbb{R}^{4\times 8}, \ \boldsymbol{b}_1 = \begin{pmatrix} x_0 \\ x_0 \\ x_0 \\ x_0 \end{pmatrix},$$

$$\boldsymbol{A}_2 = \frac{1}{4} \cdot \frac{C^2}{\epsilon^2\rho^{(2)}(x_0)}(-2, 1, 2, -1) \in \mathbb{R}^{1\times 4}, \ \boldsymbol{b}_2 = 0,$$

which exactly expresses $\phi^2_{\boldsymbol{\theta}}(x_1, x_2, x_3, x_4)$. It holds that $W = 2^3$, $L = 3$ and $B_{\boldsymbol{\theta}} = C'\epsilon^{-2}$.

Step 4: General recurrence ($d = 2^s$). The recursive construction extends naturally to arbitrary dimensions $d = 2^s$. Suppose the approximation of $x_1x_2\cdots x_{2^{s-1}}$ admits an error bound of the form

$$\|x_1x_2\cdots x_{2^{s-1}} - \phi^{s-1}_{\boldsymbol{\theta}}(x_1, \ldots, x_{2^{s-1}})\|_{W^{1,\infty}([0,1]^{2^{s-1}})} \le E_{s-1}\epsilon,$$

and define

$$\phi^s_{\boldsymbol{\theta}}(x_1, \ldots, x_{2^s}) := \phi^1_{\boldsymbol{\theta}}\big(\phi^{s-1}_{\boldsymbol{\theta}}(x_1, \ldots, x_{2^{s-1}}), \phi^{s-1}_{\boldsymbol{\theta}}(x_{2^{s-1}+1}, \ldots, x_{2^s})\big).$$

Then, similar to the analysis in Step 2, we could obtain

$$\|x_1x_2\cdots x_{2^s} - \phi^s_{\boldsymbol{\theta}}(x_1, \ldots, x_{2^s})\|_{W^{1,\infty}([0,1]^{2^s})} \le \left[(E_{s-1}+1)^2 + E_{s-1}\right]\epsilon$$

This yields a recursive relation for the approximation constants:

$$E_s = (E_{s-1}+1)^2 + E_{s-1}, \quad \text{for } 2 \le s \le \kappa.$$

To establish an explicit bound for E_κ, we set $J_s := E_s + 1$, yielding $J_s = J_{s-1}(J_{s-1}+1) \le 2J^2_{s-1}$. Letting $D_s := \log_2 J_s$, the recurrence becomes $D_s \le 1 + 2D_{s-1}$ with $D_1 = 1$. This

solves to $D_s \leq 2^s - 1$, giving $E_s \leq 2^{2^s-1} - 1$. Since $2^\kappa = d$, $E_\kappa \leq 2^{d-1} - 1$, which means setting approximate range $B = 2^d$ is enough, and

$$\|x_1 \cdots x_d - \phi_{\boldsymbol{\theta}}^\kappa(\boldsymbol{x})\|_{W^{1,\infty}([0,1]^d)} \leq (2^{d-1} - 1)\epsilon\,, \quad \boldsymbol{x} = (x_1, \cdots, x_d) \in [0,1]^d\,.$$

By recursively applying the layer-merging strategy introduced in the approximation of $x_1x_2x_3x_4$, we can construct a neural network $\phi_{\boldsymbol{\theta}}^\kappa$ to approximate the product $x_1 \cdots x_d$ which integrates intermediate linear layers. The resulting network satisfies

$$W(\phi_{\boldsymbol{\theta}}^\kappa) = 2^{\kappa+1}, \quad L(\phi_{\boldsymbol{\theta}}^\kappa) = \kappa + 1, \quad \text{and} \quad B_{\boldsymbol{\theta}}(\phi_{\boldsymbol{\theta}}^\kappa) = C'\epsilon^{-2}\,,$$

and is parameterized by $((\boldsymbol{A}_0, \boldsymbol{b}_0), \ldots, (\boldsymbol{A}_\ell, \boldsymbol{b}_\ell), \ldots, (\boldsymbol{A}_\kappa, \boldsymbol{b}_\kappa))$:

- Layer $\ell = 0$.

$$\boldsymbol{A}_0 = \begin{pmatrix} -\frac{\epsilon}{C} & -\frac{\epsilon}{C} & \cdots & 0 & 0 \\ -\frac{2\epsilon}{C} & -\frac{2\epsilon}{C} & \cdots & 0 & 0 \\ -\frac{\epsilon}{C} & \frac{\epsilon}{C} & \cdots & 0 & 0 \\ -\frac{2\epsilon}{C} & \frac{2\epsilon}{C} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\frac{\epsilon}{C} & -\frac{\epsilon}{C} \\ 0 & 0 & \cdots & -\frac{2\epsilon}{C} & -\frac{2\epsilon}{C} \\ 0 & 0 & \cdots & -\frac{\epsilon}{C} & \frac{\epsilon}{C} \\ 0 & 0 & \cdots & -\frac{2\epsilon}{C} & \frac{2\epsilon}{C} \end{pmatrix} \in \mathbb{R}^{2^{\kappa+1} \times 2^\kappa}, \quad \boldsymbol{b}_0 = \begin{pmatrix} x_0 \\ x_0 \\ x_0 \\ x_0 \\ \vdots \\ x_0 \\ x_0 \\ x_0 \\ x_0 \end{pmatrix} \in \mathbb{R}^{2^{\kappa+1}}.$$

- Layers $1 \leq \ell \leq \kappa - 1$.

$$\boldsymbol{A}_\ell = \frac{C}{4\epsilon\rho^{(2)}(x_0)} \begin{pmatrix} 2 & -1 & -2 & 1 & 2 & -1 & -2 & 1 & & & & & & & & \\ 4 & -2 & -4 & 2 & 4 & -2 & -4 & 2 & & & & & & & & \\ 2 & -1 & -2 & 1 & -2 & 1 & 2 & -1 & & & & & & & & \\ 4 & -2 & -4 & 2 & -4 & 2 & 4 & -2 & & & & & & & & \\ & & & & & & & & \ddots & & & & & & & \\ & & & & & & & & & 2 & -1 & -2 & 1 & 2 & -1 & -2 & 1 \\ & & & & & & & & & 4 & -2 & -4 & 2 & 4 & -2 & -4 & 2 \\ & & & & & & & & & 2 & -1 & -2 & 1 & -2 & 1 & 2 & -1 \\ & & & & & & & & & 4 & -2 & -4 & 2 & -4 & 2 & 4 & -2 \end{pmatrix}, \quad (35)$$

$$\boldsymbol{A}_\ell \in \mathbb{R}^{(2^{\kappa-\ell+1}) \times (2^{\kappa-\ell+2})}, \qquad \boldsymbol{b}_\ell = (x_0, \cdots, x_0)^{\mathrm{T}} \in \mathbb{R}^{2^{\kappa-\ell+1} \times 1}.$$

- Final layer $\ell = \kappa$.

$$\boldsymbol{A}_\kappa = \frac{1}{4} \cdot \frac{C^2}{\epsilon^2\rho^{(2)}(x_0)}(-2, 1, 2, -1) \in \mathbb{R}^{1\times 4}, \qquad \boldsymbol{b}_\kappa = 0 \in \mathbb{R}\,. \quad (36)$$

Step 5: Extension to arbitrary d. For arbitrary $d \geq 2$, we set $\kappa = \lceil \log_2 d \rceil$ so that $2^{\kappa-1} < d \leq 2^\kappa < 2d$. The target function is defined through dimension padding

$$\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) := \phi_{\boldsymbol{\theta}}^{\kappa}\left(\begin{pmatrix} \boldsymbol{I}_d \\ \mathbf{0}_{(2^\kappa - d)\times d} \end{pmatrix}\boldsymbol{x} + \begin{pmatrix} \mathbf{0}_{d\times 1} \\ \mathbf{1}_{(2^\kappa - d)\times 1} \end{pmatrix}\right),$$

where $\boldsymbol{I}_d$ is the $d \times d$ identity matrix, $\mathbf{0}_{p\times q}$ is the $p \times q$ zero matrix, and $\mathbf{1}_{(2^\kappa-d)\times 1}$ is the all-ones vector. This construction preserves the approximation quality:

$$\|x_1 \cdots x_d - \phi_{\boldsymbol{\theta}}(\boldsymbol{x})\|_{W^{k,\infty}([0,1]^d)} = \|x_1 \cdots x_{2^\kappa} - \phi_{\boldsymbol{\theta}}(\boldsymbol{x})\|_{W^{1,\infty}([0,1]^d)} \leq 4^d \epsilon\,,$$

while maintaining $W = 2^{\kappa+1}$ and $L = \kappa + 1$. With $C(d) = 4^d C'$, it follows that setting $B_{\boldsymbol{\theta}} = C(d)\epsilon^{-2}$ completes the proof of the lemma. ∎

Throughout the following proof, approximations are carried out on the domain Ω, a bounded open subset strictly contained in $[0,1]^d$ where the target PDE solution is defined. The function to be approximated is given by

$$f_N = \sum_{\boldsymbol{m}\in\{0,\ldots,N\}^d} \sum_{\|\boldsymbol{\alpha}\|_1 \leq n-1} c_{f,\boldsymbol{m},\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{m}}^s(\boldsymbol{x})\, \boldsymbol{x}^{\boldsymbol{\alpha}}.$$

To approximate f_N, we first construct neural networks $\phi_{\boldsymbol{\theta}}^{\boldsymbol{m},\boldsymbol{\alpha}}$ for each term $\Psi_{\boldsymbol{m}}^s(\boldsymbol{x})\,\boldsymbol{x}^{\boldsymbol{\alpha}}$.

Lemma 6 *Let $d, N, s \geq 1$, $n \geq 2$. Let $C = C(d) > 0$. For any $0 < \epsilon < \epsilon^*$, where $\epsilon^* > 0$ is sufficiently small, there exists $\phi_{\boldsymbol{\theta}}^{m,\alpha} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ where*

$$W = 2^{\lceil \log_2(d+\|\boldsymbol{\alpha}\|_1)\rceil + 1}\,,\ L = \lceil \log_2(d + \|\boldsymbol{\alpha}\|_1)\rceil + 2\,,\ B_{\boldsymbol{\theta}} = \max\{3Ns, (3d+3/2)s, C\epsilon^{-2}\}\,,$$

such that for some constant $C(n,d) > 0$,

$$\|\Psi_{\boldsymbol{m}}^s\,\boldsymbol{x}^{\boldsymbol{\alpha}} - \phi_{\boldsymbol{\theta}}^{m,\alpha}(\boldsymbol{x})\|_{W^{1,\infty}(\Omega)} \leq C(n,d)sN\epsilon\,, \quad \boldsymbol{x} \in \Omega\,,$$

for all $\boldsymbol{m} \in \{0,\cdots,N\}^d$ and $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \leq n-1$.

Proof We recall from Equation 25 that $\Psi_{\boldsymbol{m}}^s(\boldsymbol{x}) := \prod_{l=1}^d \psi^s(3N(x_l - N^{-1}m_l))$ with $\psi^s(x) = 2^{-1}[\rho(s(x+3/2)) - \rho(s(x-3/2))]$. Our goal is to approximate the function

$$\Psi_{\boldsymbol{m}}^s\,\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{l=1}^d \psi_l^s \boldsymbol{x}^{\boldsymbol{\alpha}}\,, \quad \text{where } \ \psi_l^s(x_l) = \psi^s(3N(x_l - N^{-1}m_l))\,,\ \boldsymbol{x} = (x_1,\cdots,x_d)^\top \in \Omega\,.$$

Let $\kappa = \lceil \log_2(d + \|\boldsymbol{\alpha}\|_1)\rceil$. Our construction strategy treats the target function as a product of 2^κ components. We achieve this by representing the function through an extended vector and then applying a product approximation network.

Step 1: Approximation of product factors via extended vector. We start with defining an extended vector as

$$\bar{\boldsymbol{x}} = (\psi_1^s, \cdots, \psi_d^s, \underbrace{x_{i_1}, \cdots, x_{i_{\|\boldsymbol{\alpha}\|_1}}}_{\|\boldsymbol{\alpha}\|_1}, \underbrace{1, \cdots, 1}_{2^\kappa - (d+\|\boldsymbol{\alpha}\|_1)})^\top \in [0,1]^{2^\kappa}\,,$$

where the x_{i_j} terms correspond to the variables appearing in the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$. We construct a neural network $\phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})$ to approximate $\bar{\boldsymbol{x}}$ through three distinct components.

First, for each basis function $\psi_l^s(x_l)$ with $l = 1, \ldots, d$, we achieve exact representation using two neurons with weight $3Ns$ and biases $-3m_l s \pm 3s/2$.

Second, for the $\|\boldsymbol{\alpha}\|_1$ components in the monomial term, we apply the approximation network from Lemma 2 (Equation 27). Each x_{i_j} is approximated with error ϵ using weights of magnitude $C\epsilon^{-1}$ with $C = C(d) > 0$ from Lemma 5, and bias x_0. We denote these approximated values as $\hat{x}_i$ and define the approximated monomial as $\hat{\boldsymbol{x}}^{\boldsymbol{\alpha}}$. Since Ω is strictly contained in $[0,1]^d$, choosing ϵ sufficiently small ensures that $\hat{\boldsymbol{x}} \in [0,1]^d$.

Finally, the remaining $2^\kappa - (d + \|\boldsymbol{\alpha}\|_1)$ components are implemented as constant functions with value 1, realized by neurons with zero input weights and unit bias.

The resulting network output is

$$\phi_{\boldsymbol{\theta}_*}(\boldsymbol{x}) = (\psi_1^s, \cdots, \psi_d^s, \underbrace{\hat{x}_{i_1}, \cdots, \hat{x}_{i_{\|\boldsymbol{\alpha}\|_1}}}_{\|\boldsymbol{\alpha}\|_1}, \underbrace{1, \cdots, 1}_{2^\kappa - (d+\|\boldsymbol{\alpha}\|_1)})^\top \in [0,1]^{2^\kappa}.$$

For notational convenience, denote by $P(\boldsymbol{x}) := \prod_{i=1}^d x_i$ for a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$. It is easy to check that $\|\psi_l^s\|_{W^{1,\infty}(\Omega)} = \mathcal{O}(sN)$. Thus, repeatedly applying the product rule (Equation 30) from Lemma 3 yields:

$$\begin{aligned}
&\left\|\Psi_{\boldsymbol{m}}^s(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}} - P \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\right\|_{W^{1,\infty}(\Omega)} \\
&\qquad = \left\| \prod_{l=1}^d \psi_l^s(x_l)\boldsymbol{x}^{\boldsymbol{\alpha}} - \prod_{l=1}^d \psi_l^s(x_l)\hat{\boldsymbol{x}}^{\boldsymbol{\alpha}} \right\|_{W^{1,\infty}(\Omega)} \leq C_1(n,d)sN\epsilon\,.
\end{aligned}$$

Step 2: Product network composition and parameterization. Letting $C = C(d) > 0$ from Lemma 5, we construct a neural network $\phi_{\tilde{\boldsymbol{\theta}}}^\kappa$ that approximates the product function P on $[0,1]^{2^\kappa}$ with parameters $W(\phi_{\tilde{\boldsymbol{\theta}}}^\kappa) = 2^{\kappa+1}$, $L(\phi_{\tilde{\boldsymbol{\theta}}}^\kappa) = \kappa + 1$, $B_{\boldsymbol{\theta}}(\phi_{\tilde{\boldsymbol{\theta}}}^\kappa) = C\epsilon^{-2}$. Applying the chain rule from Lemma 3, the approximation error satisfies

$$\begin{aligned}
\left\|P \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x}) - \phi_{\tilde{\boldsymbol{\theta}}}^\kappa \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\right\|_{W^{1,\infty}(\Omega)} &= \left\|(P - \phi_{\tilde{\boldsymbol{\theta}}}^\kappa) \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\right\|_{W^{1,\infty}(\Omega)} \\
&\leq C \max\left\{\|P - \phi_{\tilde{\boldsymbol{\theta}}}^\kappa\|_{L^\infty([0,1]^\kappa)},\ |P - \phi_{\tilde{\boldsymbol{\theta}}}^\kappa|_{W^{1,\infty}([0,1]^\kappa)} \cdot |\phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})|_{W^{1,\infty}(\Omega;\,[0,1]^\kappa)}\right\} \\
&\leq C_2(n,d)sN\epsilon\,.
\end{aligned}$$

We define our final network as the composition $\phi_{\boldsymbol{\theta}}^{\boldsymbol{m},\boldsymbol{\alpha}} := \phi_{\tilde{\boldsymbol{\theta}}}^\kappa \circ \phi_{\boldsymbol{\theta}_*}$ with parameter

$$((\boldsymbol{A}_*, \boldsymbol{b}_*), (\boldsymbol{A}_0, \boldsymbol{b}_0), (\boldsymbol{A}_1, \boldsymbol{b}_1), \cdots, (\boldsymbol{A}_\ell, \boldsymbol{b}_\ell), \cdots, (\boldsymbol{A}_\kappa, \boldsymbol{b}_\kappa))\,.$$

Below, we provide an explicit construction of the parameter matrix. To simplify the presentation, we assume here that d and $\|\boldsymbol{\alpha}\|_1$ are both even with $\|\boldsymbol{\alpha}\|_1 = 2$ and $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1 x_2$; the construction for the odd case is slightly different but conceptually equivalent.

- For the first layer implementing $\phi_{\boldsymbol{\theta}_*}$, denote by $\boldsymbol{\tau}_N := (3Ns, 3Ns)^{\mathrm{T}} \in \mathbb{R}^2$, $\boldsymbol{b}_i := (-3m_i s + 3/2s, -3m_i s - 3/2s)^{\mathrm{T}} \in \mathbb{R}^2$, $\mathbf{1}_n \in \mathbb{R}^n$ the all-ones vector, and $\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$

the zero matrix. Then, we have

$$\boldsymbol{A}_* = \begin{pmatrix} \boldsymbol{\tau}_N & & & \\ & \boldsymbol{\tau}_N & & \\ & & \ddots & \\ & & & \boldsymbol{\tau}_N \\ -\frac{\epsilon}{C} & 0 & \cdots & 0 \\ 0 & -\frac{\epsilon}{C} & \cdots & 0 \\ \multicolumn{4}{c}{\mathbf{0}_{(2^\kappa - \|\boldsymbol{\alpha}\|_1)\times d}} \end{pmatrix} \in \mathbb{R}^{(d+2^\kappa)\times d}, \qquad \boldsymbol{b}_* = \begin{pmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_d \\ x_0 \mathbf{1}_{\|\boldsymbol{\alpha}\|_1} \\ \mathbf{1}_{2^\kappa - \|\boldsymbol{\alpha}\|_1} \end{pmatrix} \in \mathbb{R}^{d+2^\kappa}.$$

- Next, denote by

$$\boldsymbol{\Pi}^1 : \begin{pmatrix} \frac{-\epsilon}{2C} & \frac{\epsilon}{2C} & \frac{-\epsilon}{2C} & \frac{\epsilon}{2C} \\ \frac{-\epsilon}{C} & \frac{\epsilon}{C} & \frac{-\epsilon}{C} & \frac{\epsilon}{C} \\ \frac{-\epsilon}{2C} & \frac{\epsilon}{2C} & \frac{\epsilon}{2C} & \frac{-\epsilon}{2C} \\ \frac{-\epsilon}{C} & \frac{\epsilon}{C} & \frac{\epsilon}{C} & \frac{-\epsilon}{C} \end{pmatrix}, \quad \boldsymbol{\Pi}^2 : \begin{pmatrix} \frac{1}{\rho^{(1)}(x_0)} & \frac{1}{\rho^{(1)}(x_0)} \\ \frac{2}{\rho^{(1)}(x_0)} & \frac{2}{\rho^{(1)}(x_0)} \\ \frac{1}{\rho^{(1)}(x_0)} & -\frac{1}{\rho^{(1)}(x_0)} \\ \frac{2}{\rho^{(1)}(x_0)} & -\frac{2}{\rho^{(1)}(x_0)} \end{pmatrix}, \quad \boldsymbol{\Pi}^3 : \begin{pmatrix} \frac{-\epsilon}{C\rho(1)} & \frac{-\epsilon}{C\rho(1)} \\ \frac{-2\epsilon}{C\rho(1)} & \frac{-2\epsilon}{C\rho(1)} \\ \frac{-\epsilon}{C\rho(1)} & \frac{\epsilon}{C\rho(1)} \\ \frac{-2\epsilon}{C\rho(1)} & \frac{2\epsilon}{C\rho(1)} \end{pmatrix},$$

and $\boldsymbol{\pi} := (x_0 - 2\rho(x_0)/\rho^{(1)}(x_0), x_0 - 4\rho(x_0)/\rho^{(1)}(x_0), x_0, x_0) \in \mathbb{R}^{1\times 4}$. Using the merging technique, the weight matrix for $\ell = 0$ is as follows

$$\begin{aligned} \boldsymbol{A}_0 &= \operatorname{diag}\big(\underbrace{\boldsymbol{\Pi}^1, \ldots, \boldsymbol{\Pi}^1}_{d/2}, \underbrace{\boldsymbol{\Pi}^2, \ldots, \boldsymbol{\Pi}^2}_{\|\boldsymbol{\alpha}\|_1/2}, \underbrace{\boldsymbol{\Pi}^3, \ldots, \boldsymbol{\Pi}^3}_{(2^\kappa - d - \|\boldsymbol{\alpha}\|_1)/2}\big) \in \mathbb{R}^{(2^{\kappa+1})\times(d+2^\kappa)}, \\ \boldsymbol{b}_0 &= \big(x_0 \mathbf{1}_{2d}^{\mathrm{T}}, \underbrace{\boldsymbol{\pi}, \ldots, \boldsymbol{\pi}}_{\|\boldsymbol{\alpha}\|_1/2}, x_0 \mathbf{1}_{2^{\kappa+1} - 2d - 2\|\boldsymbol{\alpha}\|_1}^{\mathrm{T}}\big)^{\mathrm{T}} \in \mathbb{R}^{2^{\kappa+1}}. \end{aligned}$$

- For layers $\ell = 1, \ldots, \kappa$, the construction mirrors Lemma 5, using the parameters from Equations (35) and (36) to form the product approximation network $\phi^\kappa_{\tilde{\boldsymbol{\theta}}}$.

Step 3: Final architecture and error bound. The composed network has the following parameters: $W = 2^{\kappa+1} = 2^{\lceil \log_2(d+\|\boldsymbol{\alpha}\|_1)\rceil + 1}$, $L = \kappa + 2 = \lceil \log_2(d + \|\boldsymbol{\alpha}\|_1)\rceil + 1$ and $B_{\boldsymbol{\theta}} = \max\{3Ns, (3d+3/2)s, C\epsilon^{-2}\}$. The final approximation error is bounded by the triangle inequality as

$$\begin{aligned} \|\Psi^s_{\boldsymbol{m}} \boldsymbol{x}^{\boldsymbol{\alpha}} - \phi^{m,\alpha}_{\boldsymbol{\theta}}\|_{W^{1,\infty}(\Omega)} &= \|\Psi^s_{\boldsymbol{m}}(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}} - \phi^\kappa_{\tilde{\boldsymbol{\theta}}} \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\|_{W^{1,\infty}(\Omega)} \\ &\le \big\|\Psi^s_{\boldsymbol{m}}(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}} - P \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\big\|_{W^{1,\infty}(\Omega)} + \big\|P \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x}) - \phi^\kappa_{\tilde{\boldsymbol{\theta}}} \circ \phi_{\boldsymbol{\theta}_*}(\boldsymbol{x})\big\|_{W^{1,\infty}(\Omega)} \\ &\le C_3(n,d) s N \epsilon, \end{aligned}$$

which completes the proof. ∎

Now, we can construct the parallel neural network $\Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ to approximate f_N.

Theorem 6 *For some sufficiently small $\epsilon^* > 0$ and any $0 < \epsilon < \epsilon^*$, there exists a neural network $\Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ with*

$$\bar{\mathfrak{m}} = C_1(n,d)(N+1)^d\,, \quad \bar{M} = C_2(n,d)N^{d/p}(N+1)\,, \quad \bar{W} = 2^{\lceil \log_2(d+n-1)\rceil+1}\,,$$
$$\bar{L} = \lceil \log_2(d+n-1)\rceil + 2\,, \quad B_{\bar{\boldsymbol{\theta}}} = \max\{3Ns, (3d+3/2)s, C(d)\epsilon^{-2}\}\,,$$

such that

$$\|f_N(\boldsymbol{x}) - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\|_{W^{1,\infty}(\Omega)} \le C_3(n,d)sN(N+1)^d\epsilon\,, \quad \boldsymbol{x} = (x_1, \cdots, x_d)^\top \in \Omega\,.$$

Proof By Lemma 6, for any $\boldsymbol{m} \in \{0, \cdots, N\}^d$ and $\|\boldsymbol{\alpha}\|_1 \le n-1$, $\Psi_{\boldsymbol{m}}^s \boldsymbol{x}^{\boldsymbol{\alpha}}$ can be approximated by $\phi_{\bar{\boldsymbol{\theta}}}^{\boldsymbol{m},\boldsymbol{\alpha}} \in \mathcal{NN}(\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}})$ with $\bar{W} = 2^{\lceil \log_2(d+n-1)\rceil+1}$, $\bar{L} = \lceil \log_2(d+n-1)\rceil + 2$ and $B_{\bar{\boldsymbol{\theta}}} = \max\{3Ns, (3d+3/2)s, C(d)\epsilon^{-2}\}$. By Proposition 4, it holds that

$$|c_{f,\boldsymbol{m},\boldsymbol{\alpha}}| \le C_{\text{Prop }4}\|\tilde{f}\|_{W^{n,p}(\Omega_{\boldsymbol{m},N})}N^{d/p} \le C_{\text{Prop }4}\|f\|_{W^{n,p}(\Omega)}N^{d/p} \le C_{\text{Prop }4}N^{d/p}\,.$$

Also, observe that

$$\sum_{\|\boldsymbol{\alpha}\|_1 \le n-1} 1 = \sum_{j=0}^{n-1} \sum_{\|\boldsymbol{\alpha}\|_1 = j} 1 \le \sum_{j=0}^{n-1} d^j \le nd^{n-1}.$$

Let $\Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$, that is,

$$\Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(\boldsymbol{x}) = \sum_{k=1}^{\bar{\mathfrak{m}}} c_k \phi_{\bar{\boldsymbol{\theta}}}^k(\boldsymbol{x}) := \sum_{\boldsymbol{m}\in\{0,\ldots,N\}^d} \sum_{\|\boldsymbol{\alpha}\|_1 \le n-1} c_{f,\boldsymbol{m},\boldsymbol{\alpha}} \phi_{\bar{\boldsymbol{\theta}}}^{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{x})\,,$$

where $\bar{\mathfrak{m}} = (N+1)^d n\, d^{n-1}$ and $\bar{M} = C_{\text{Prop }4}N^{d/p}(N+1)^d n\, d^{n-1}$. Note that here, any surplus c_k coefficients and $\phi_{\bar{\boldsymbol{\theta}}}^k$ sub-networks are set to zero. Then, it holds that

$$\begin{aligned}
\left\|f_N(\boldsymbol{x}) - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(\boldsymbol{x})\right\|_{W^{1,p}(\Omega)} &= \left\|f_N(\boldsymbol{x}) - \sum_{\boldsymbol{m}\in\{0,\ldots,N\}^d} \sum_{\|\boldsymbol{\alpha}\|_1 \le n-1} c_{f,\boldsymbol{m},\boldsymbol{\alpha}}\, \phi_{\bar{\boldsymbol{\theta}}}^{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{x})\right\|_{W^{1,p}(\Omega)} \\
&\le \sum_{\boldsymbol{m}\in\{0,\ldots,N\}^d} \sum_{\|\boldsymbol{\alpha}\|_1 \le n-1} |c_{f,\boldsymbol{m},\boldsymbol{\alpha}}| \cdot \left\|\Psi_{\boldsymbol{m}}^s \boldsymbol{x}^{\boldsymbol{\alpha}} - \phi_{\bar{\boldsymbol{\theta}}}^{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{x})\right\|_{W^{1,p}(\Omega)} \\
&\le C_{\text{Prop }4}C_{\text{Lem }6}(n,d)N^{d/p}2^d\, n\, d^{n-1}(N+1)^d sN\epsilon\,.
\end{aligned}$$

When $p = \infty$, we have $\left\|f_N(\boldsymbol{x}) - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(\boldsymbol{x})\right\|_{W^{1,\infty}(\Omega)} \le C(n,d)sN(N+1)^d\epsilon$. ■

A.4 Approximation Error Bound for Parallel Neural Networks

Finally, by combining Proposition 4 with Theorem 6, we can derive an upper bound for the approximation error.

Theorem 7 *Let $n, d, \bar{\mathfrak{m}} \in \mathbb{N}$ and $1 \le p \le \infty$. Let the target function $f \in \mathcal{F}_{n,d,p}$. For some sufficiently small $\epsilon^* > 0$ and any $0 < \epsilon < \epsilon^*$, there exists $\Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$*

with $\bar{\mathfrak{m}} = C_1(n,d)\epsilon^{-\frac{d}{n-\mu-1}}$, $\bar{M} = C_2(n,d)\epsilon^{-\frac{d(p+1)}{(n-\mu-1)p}}$, $\bar{W} = 2^{\lceil \log_2(d+n-1)\rceil+1}$, $\bar{L} = \lceil \log_2(d+n-1)\rceil + 2$, *and* $B_{\bar{\boldsymbol{\theta}}} = C_3(n,d)\epsilon^{-\frac{2d+2n}{n-\mu-1}}$ *such that*

$$\left\|f - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\right\|_{W^{1,p}(\Omega)} \leq \epsilon\,,$$

where μ *is an arbitrarily small positive number.*

Proof First, by Proposition 4, we choose

$$N = \left\lceil (\epsilon/2C_{\text{Prop 4}})^{-1/(n-\mu-1)} \right\rceil \quad \text{and} \quad s = N^{\mu}\,.$$

Then, for each $\boldsymbol{m} \in \{0,\cdots,N\}^d$, there exists a polynomial $p_{\boldsymbol{m}}(\boldsymbol{x}) = \sum_{\|\boldsymbol{\alpha}\|_1 \leq n-1} c_{f,\boldsymbol{m},\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$ such that

$$\left\|f - \sum_{\boldsymbol{m}\in\{0,\cdots,N\}^d} \Psi_{\boldsymbol{m}}^s p_{\boldsymbol{m}}\right\|_{W^{1,p}(\Omega)} \leq C_{\text{Prop 4}}\left(\frac{1}{N}\right)^{n-\mu-1} \leq C_{\text{Prop 4}} \cdot \frac{\epsilon}{2C_{\text{Prop 4}}} = \frac{\epsilon}{2}\,.$$

Secondly, since $\|u\|_{W^{1,p}(\Omega)} \leq C(\Omega)\|u\|_{W^{1,\infty}(\Omega)}$, we apply Theorem 6 with N and s chosen above, and set $\epsilon_{\text{Thm 6}} = C'(n,d)\cdot\epsilon^{1+\frac{d+\mu+1}{n-\mu-1}}$, then it holds that

$$\left\|f_N(\boldsymbol{x}) - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(\boldsymbol{x})\right\|_{W^{1,\infty}(\Omega)} \leq C(n,d)(N+1)^d sN\epsilon_{\text{Thm 6}} \leq (2C(\Omega))^{-1}\epsilon\,.$$

Hence, the total approximation error satisfies

$$\left\|f - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\right\|_{W^{1,p}(\Omega)} \leq \left\|f - f_N\right\|_{W^{1,p}(\Omega)} + \left\|f_N - \Phi_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\right\|_{W^{1,p}(\Omega)} \leq \epsilon\,,$$

with $\bar{\mathfrak{m}} = C_1(n,d)\epsilon^{-\frac{d}{n-\mu-1}}$, $\bar{M} = C_2(n,d)\epsilon^{-\frac{d(p+1)}{(n-\mu-1)p}}$, $\bar{W} = 2^{\lceil \log_2(d+n-1)\rceil+1}$, $\bar{L} = \lceil \log_2(d+n-1)\rceil + 2$, and $B_{\bar{\boldsymbol{\theta}}} = C_3(n,d)\epsilon^{-\frac{2d+2n}{n-\mu-1}}$. ■

Appendix B. Detailed Statistical Error Analysis

In this section, we provide a detailed expansion of Section 3.4 in the main text, which is mainly based on the approach in Kohler and Krzyzak (2023a) and Jiao et al. (2023a). Recall the definition of statistical error in Section 3.4:

$$\mathcal{E}_{sta} := \sup_{u\in\mathcal{PNN}} \left|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)\right|.$$

The task is to control $\mathcal{E}_{sta}$ with high probability. To achieve this goal, we organize the analysis into following four parts.

B.1 Some Neural Network Function Classes

We begin with some auxiliary function classes that are essential for subsequent analysis. First, we define the squared classes with respect to $\mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}})$:

$$\mathcal{F}_1' := \left\{ \pm f : \Omega \to \mathbb{R} \mid \exists\ u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}})\ \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = \left[\partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\right]^2 \right\},$$

$$\mathcal{F}_2' := \left\{ \pm f : \Omega \to \mathbb{R} \mid \exists\, u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}}) \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = u^2_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \right\}.$$

Further, denote by

$$\begin{aligned}
\mathcal{F}_1 &:= \{ f : \Omega \to \mathbb{R} \mid \exists\, u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}}) \ \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = \partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \}, \\
\mathcal{F}_2 &:= \{ f : \Omega \to \mathbb{R} \mid \exists\, u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}}) \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \}, \\
\mathcal{F}_3 &:= \{ f : \partial\Omega \to \mathbb{R} \mid \exists\, u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, W, L, B_{\boldsymbol{\theta}}) \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})|_{\partial\Omega} \}.
\end{aligned}$$

Finally, we define the sub-network function classes:

$$\begin{aligned}
\mathcal{F}_{1,sub} &:= \{ f : \Omega \to \mathbb{R} \mid \exists\, \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}}) \ \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = \partial_{x_1} \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \}, \\
\mathcal{F}_{2,sub} &:= \{ f : \Omega \to \mathbb{R} \mid \exists\, \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}}) \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \}, \\
\mathcal{F}_{3,sub} &:= \{ f : \partial\Omega \to \mathbb{R} \mid \exists\, \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}}) \ s.t.\ f(\boldsymbol{x};\boldsymbol{\theta}) = \phi_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\partial\Omega} \}.
\end{aligned}$$

Lemmas 7–9 establish several regularity properties for $\mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ and the newly defined $\mathcal{F}_{i,sub}$. Proofs of Lemmas 7 and 8 can be found in Appendices B.5 and B.6. Below, we assume that $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ are two arbitrary parameter vectors satisfying

$$\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \in \Theta = [-B_{\boldsymbol{\theta}}, B_{\boldsymbol{\theta}}]^{\mathfrak{D}(W,L,d)}, \quad B_{\boldsymbol{\theta}} \geq 1.$$

Lemma 7 *For any $\phi_{\boldsymbol{\theta}} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ and $\boldsymbol{x} \in \Omega$, we have $|\phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq (W+1)B_{\boldsymbol{\theta}}$. Moreover, for any $\phi_{\boldsymbol{\theta}}, \phi_{\tilde{\boldsymbol{\theta}}} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ and $\boldsymbol{x} \in \Omega$, it holds that*

$$|\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) - \phi_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x})| \leq 2W^L \sqrt{L} B_{\boldsymbol{\theta}}^{L-1} \left\| \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right\|_2.$$

Lemma 8 *For any $\phi_{\boldsymbol{\theta}} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$ and $\boldsymbol{x} \in \Omega$, we have $|\partial_{x_m} \phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq W^{L-1} B_{\boldsymbol{\theta}}^L$. Moreover, for any $\phi_{\boldsymbol{\theta}}, \phi_{\tilde{\boldsymbol{\theta}}} \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$, $\boldsymbol{x} \in \Omega$ and $m \in [d]$, it holds that*

$$|\partial_{x_m} \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) - \partial_{x_m} \phi_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x})| \leq 2W^{2L-1} \sqrt{L}(L+1) B_{\boldsymbol{\theta}}^{2L} \left\| \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right\|_2, \quad \forall \boldsymbol{x} \in \Omega.$$

Lemma 9 is a direct result of Lemma 7 and Lemma 8.

Lemma 9 *Let $f_i(\cdot;\boldsymbol{\theta}), f_i(\cdot;\tilde{\boldsymbol{\theta}}) \in \mathcal{F}_{i,sub}$, $i = 1, 2, 3$. Then for any $\boldsymbol{x} \in \Omega$, they satisfy both boundedness and Lipschitz conditions with respect to parameters:*

$$\left| f_i(\boldsymbol{x};\boldsymbol{\theta}) \right| \leq B_i, \quad \left| f_i(\boldsymbol{x};\boldsymbol{\theta}) - f_i(\boldsymbol{x};\tilde{\boldsymbol{\theta}}) \right| \leq L_i \left\| \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right\|_2,$$

where the constants are given by:

$$\begin{aligned}
B_1 &= W^{L-1} B_{\boldsymbol{\theta}}^L, & B_2 &= B_3 = (W+1) B_{\boldsymbol{\theta}}, \\
L_1 &= 2W^{2L-1} \sqrt{L}(L+1) B_{\boldsymbol{\theta}}^{2L}, & L_2 &= L_3 = 2W^L \sqrt{L} B_{\boldsymbol{\theta}}^{L-1}.
\end{aligned}$$

B.2 Controlling Statistical Error Through Rademacher Complexity

Then, we focus on the expectation of $\mathcal{E}_{sta}$ with respect to the Monte Carlo samples, $\mathbb{E}[\mathcal{E}_{sta}]$. The following lemma is direct.

Lemma 10 *Let $\mathcal{D} := \{X_p\}_{p=1}^{N_{\text{in}}} \cup \{Y_q\}_{q=1}^{N_b}$. The expected statistical error decomposes as:*

$$\begin{aligned}\mathbb{E}[\mathcal{E}_{sta}] &= \mathbb{E}_{\mathcal{D}}\left[\sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}}\left|\mathcal{L}(u_{\mathfrak{m},\boldsymbol{\theta}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})\right|\right] \\ &\leq \sum_{i=1}^{4}\mathbb{E}_{\mathcal{D}}\left[\sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}}\left|\mathcal{L}_i(u_{\mathfrak{m},\boldsymbol{\theta}}) - \widehat{\mathcal{L}}_i(u_{\mathfrak{m},\boldsymbol{\theta}})\right|\right] =: \sum_{i=1}^{4}\mathbb{E}[\mathcal{E}_{sta}^i]\,.\end{aligned}$$

where

$$\begin{aligned}\mathcal{L}_1(u) &= \frac{|\Omega|}{2}\mathbb{E}_{X\sim U(\Omega)}\left[\|\nabla u(X)\|^2\right], & \mathcal{L}_2(u) &= \frac{|\Omega|}{2}\mathbb{E}_{X\sim U(\Omega)}\left[w(X)u^2(X)\right], \\ \mathcal{L}_3(u) &= -|\Omega|\mathbb{E}_{X\sim U(\Omega)}\left[h(X)u(X)\right], & \mathcal{L}_4(u) &= -|\partial\Omega|\mathbb{E}_{Y\sim U(\partial\Omega)}\left[g(Y)Tu(Y)\right].\end{aligned}$$

and $\widehat{\mathcal{L}}_i(u)$ is the discrete version of $\mathcal{L}_i(u)$, for example,

$$\widehat{\mathcal{L}}_1(u) = \frac{|\Omega|}{2N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\|\nabla u(X_p)\|^2\,.$$

To control $\mathbb{E}[\mathcal{E}_{sta}^i]$ using symmetrization techniques, we introduce Rademacher complexity as our analytical foundation.

Definition 4 *Two types of Rademacher complexity of function class $\mathcal{F}$ associate with random sample $\{X_k\}_{k=1}^N$ are defined as*

$$\begin{aligned}\mathfrak{R}_N(\mathcal{F}) &= \mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^N}\left[\sup_{u\in\mathcal{F}}\frac{1}{N}\sum_{k=1}^{N}\sigma_k u(X_k)\right], \\ \hat{\mathfrak{R}}_N(\mathcal{F}) &= \mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^N}\left[\sup_{u\in\mathcal{F}}\frac{1}{N}\left|\sum_{k=1}^{N}\sigma_k u(X_k)\right|\right],\end{aligned}$$

where, $\{\sigma_k\}_{k=1}^N$ are N i.i.d Rademacher variables with $\mathbb{P}(\sigma_k = 1) = \mathbb{P}(\sigma_k = -1) = \frac{1}{2}$.

For Rademacher complexity $\mathfrak{R}_N(\mathcal{F})$, we have following two structural results. Proofs of Lemmas 11 and 12 can be found in Appendices B.7 and B.8.

Lemma 11 *Let $\mathcal{F}$ be a class of functions mapping from Ω to $\mathbb{R}$. If $a : \Omega \to \mathbb{R}$ is a function such that $|a(\boldsymbol{x})| \leq \mathcal{B}$ for all $\boldsymbol{x} \in \Omega$, then*

$$\mathfrak{R}_N(a\cdot\mathcal{F}) \leq \mathcal{B}\mathfrak{R}_N(\mathcal{F})\,,$$

where $a\cdot\mathcal{F} := \{\bar{f} : \Omega\to\mathbb{R} \mid \bar{f}(\boldsymbol{x}) = a(\boldsymbol{x})f(\boldsymbol{x})$ for some $f\in\mathcal{F}\}$.

Lemma 12 *Let $\mathcal{F}$ be a class of functions mapping from Ω to $\mathbb{R}$, and let $\Phi \in \mathbb{R}$ be a λ-Lipschitz function. Then, it holds that*

$$\mathfrak{R}_N(\Phi \circ \mathcal{F}) \leq \lambda \mathfrak{R}_N(\mathcal{F}) .$$

The following lemma bounds $\mathbb{E}[\mathcal{E}_{sta}^i]$ in terms of $\hat{\mathfrak{R}}_N(\mathcal{F}_{i,sub})$.

Lemma 13 *Let $u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$. It holds that*

$$\mathbb{E}[\mathcal{E}_{sta}^1] \leq 4d|\Omega| W^{L-1} B_{\boldsymbol{\theta}}^L M^2 \hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{1,sub}) , \qquad \mathbb{E}[\mathcal{E}_{sta}^3] \leq 2B_0|\Omega| M \hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{2,sub}) ,$$
$$\mathbb{E}[\mathcal{E}_{sta}^2] \leq 4B_0|\Omega| (W+1) B_{\boldsymbol{\theta}} M^2 \hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{2,sub}) , \quad \mathbb{E}[\mathcal{E}_{sta}^4] \leq 2B_0|\partial\Omega| M \hat{\mathfrak{R}}_{N_b}(\mathcal{F}_{3,sub}) .$$

Proof We will complete the proof in the following three steps.

Step 1. Take $\{\tilde{X}_p\}_{p=1}^{N_{\text{in}}}$ as an independent copy of $\{X_p\}_{p=1}^{N_{\text{in}}}$. Denote by $\mathcal{D}_X := \{X_p\}_{p=1}^{N_{\text{in}}}$, $\mathcal{D}_{\tilde{X}} := \{\tilde{X}_p\}_{p=1}^{N_{\text{in}}}$ and $\mathcal{D}_{X,\tilde{X}} := \{X_p, \tilde{X}_p\}_{p=1}^{N_{\text{in}}}$. Then, we have

$$\mathcal{L}_1(u_{\mathfrak{m},\boldsymbol{\theta}}) - \widehat{\mathcal{L}}_1(u_{\mathfrak{m},\boldsymbol{\theta}}) = \frac{|\Omega|}{2}\left[\mathbb{E}_{X\sim U(\Omega)}\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(X)\|^2 - \frac{1}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)\|^2\right]$$
$$= \frac{|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\tilde{X}}}\left[\sum_{m=1}^{d}\sum_{p=1}^{N_{\text{in}}}\left[\left(\partial_{x_m} u_{\mathfrak{m},\boldsymbol{\theta}}(\tilde{X}_p)\right)^2 - \left(\partial_{x_m} u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)\right)^2\right]\right] .$$

Denote by $\mathcal{D}_{\sigma,X} := \mathcal{D}_X \cup \{\sigma_p\}_{p=1}^{N_{\text{in}}}$ and $\mathcal{D}_{\sigma,X,\tilde{X}} := \mathcal{D}_{X,\tilde{X}} \cup \{\sigma_p\}_{p=1}^{N_{\text{in}}}$. It holds that

$$\mathbb{E}[\mathcal{E}_{sta}^1] = \mathbb{E}_{\mathcal{D}_X}\Big[\sup_{u\in\mathcal{PNN}}\big|\mathcal{L}_1(u) - \widehat{\mathcal{L}}_1(u)\big|\Big]$$
$$= \frac{|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_X}\left[\sup_{u\in\mathcal{PNN}}\left|\mathbb{E}_{\mathcal{D}_{\tilde{X}}}\left[\sum_{m=1}^{d}\sum_{p=1}^{N_{\text{in}}}\left[\left(\partial_{x_m}u(X_p)\right)^2 - \left(\partial_{x_m}u(X_p)\right)^2\right]\right]\right|\right]$$
$$\leq \frac{|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{X,\tilde{X}}}\left[\sup_{u\in\mathcal{PNN}}\sum_{m=1}^{d}\left|\sum_{p=1}^{N_{\text{in}}}\left[\left(\partial_{x_m}u(X_p)\right)^2 - \left(\partial_{x_m}u(X_p)\right)^2\right]\right|\right]$$
$$= \frac{|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X,\tilde{X}}}\left[\sup_{u\in\mathcal{PNN}}\sum_{m=1}^{d}\left|\sum_{p=1}^{N_{\text{in}}}\sigma_p\left[\left(\partial_{x_m}u(X_p)\right)^2 - \left(\partial_{x_m}u(X_p)\right)^2\right]\right|\right]$$
$$\leq \frac{d|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X,\tilde{X}}}\left[\sup_{f\in\mathcal{F}_1'}\left|\sum_{p=1}^{N_{\text{in}}}\sigma_p f(\tilde{X}_p)\right|\right] + \frac{d|\Omega|}{2N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X,\tilde{X}}}\left[\sup_{f\in\mathcal{F}_1'}\left|\sum_{p=1}^{N_{\text{in}}}-\sigma_p f(X_p)\right|\right]$$
$$= \frac{d|\Omega|}{N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{f\in\mathcal{F}_1'}\left|\sum_{p=1}^{N_{\text{in}}}\sigma_p f(X_p)\right|\right] = \frac{d|\Omega|}{N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X}}\Big[\sup_{f\in\mathcal{F}_1'}\sum_{p=1}^{N_{\text{in}}}\sigma_p f(X_p)\Big] = d\,|\Omega|\,\mathfrak{R}_{N_{\text{in}}}(\mathcal{F}_1') ,$$

where the fifth step is due to the symmetric structure of the $\mathcal{PNN}$ class with respect to the d variable components $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ and the seventh step uses that $\mathcal{F}_1'$ contains both f and $-f$ for every function f in the class. Similarly, it holds that

$$\mathbb{E}[\mathcal{E}_{sta}^2] \leq \frac{|\Omega|}{N_{\text{in}}}\mathbb{E}_{\mathcal{D}_{\sigma,X}}\Big[\sup_{f\in\mathcal{F}_2'}\sum_{p=1}^{N_{\text{in}}}\sigma_p\, w(X_p) f(X_p)\Big] = |\Omega|\mathfrak{R}_N(w\cdot\mathcal{F}_2') \leq B_0|\Omega|\mathfrak{R}_{N_{\text{in}}}(\mathcal{F}_2') ,$$

where we use Lemma 11 in the last step. Directly, we have

$$\mathbb{E}[\mathcal{E}_{sta}^3] \le 2B_0|\Omega|\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_2)\,, \qquad \mathbb{E}[\mathcal{E}_{sta}^4] \le 2B_0|\partial\Omega|\mathfrak{R}_{N_b}(\mathcal{F}_3)\,.$$

Step 2. Since $u_{\mathfrak{m},\boldsymbol{\theta}} = \sum_{k=1}^{\mathfrak{m}} c_k\phi_{\boldsymbol{\theta}}^k$ and $\sum_k |c_k| \le M$, by Lemmas 7 and 8, we have

$$|u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})| \le M(W+1)B_{\boldsymbol{\theta}}\,, \quad |\partial_{x_m} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})| \le MW^{L-1}B_{\boldsymbol{\theta}}^L\,.$$

Define $\mathcal{F}_1''$ and $\mathcal{F}_2''$ as the non-symmetrized versions of $\mathcal{F}_1'$ and $\mathcal{F}_2'$ (thus, $\mathcal{F}_1'' = \{f^2 \mid f \in \mathcal{F}_1\}$ and $\mathcal{F}_2'' = \{f^2 \mid f \in \mathcal{F}_2\}$). Letting $\Phi(x) = x^2$, then by Lemma 12, it holds that

$$\begin{aligned}
\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_1') &\le 2\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_1'') \le 4MW^{L-1}B_{\boldsymbol{\theta}}^L\,\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_1)\,,\\
\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_2') &\le 2\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_2'') \le 4M(W+1)B_{\boldsymbol{\theta}}\,\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_2)\,.
\end{aligned}$$

Step 3. Finally, we will relate $\mathfrak{R}_N(\mathcal{F}_i)$ to $\hat{\mathfrak{R}}_N(\mathcal{F}_{i,sub})$. Taking $\mathcal{F}_1$ as an example, we have

$$\begin{aligned}
\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_1) &= \mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}} \frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p \partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)\right]\\
&= \mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}}\in\Theta^{\mathfrak{m}}} \frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p \left[\sum_{k=1}^{\mathfrak{m}} c_k\,\partial_{x_1}\phi_{\boldsymbol{\theta}}^k(X_p)\right]\right]\\
&= \mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}}\in\Theta^{\mathfrak{m}}} \sum_{k=1}^{\mathfrak{m}} c_k \left[\frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p\,\partial_{x_1}\phi_{\boldsymbol{\theta}}^k(X_p)\right]\right]\\
&\le \mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}}\in\Theta^{\mathfrak{m}}} \sum_{k=1}^{\mathfrak{m}} |c_k| \cdot \left|\frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p\partial_{x_1}\phi_{\boldsymbol{\theta}}^k(X_p)\right|\right]\\
&\le \mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}}\in\Theta^{\mathfrak{m}}} \sum_{k=1}^{\mathfrak{m}} |c_k| \cdot \sup_{k\in\{1,\ldots,\mathfrak{m}\}}\left|\frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p\,\partial_{x_1}\phi_{\boldsymbol{\theta}}^k(X_p)\right|\right]\\
&\le M\,\mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}}\in\Theta^{\mathfrak{m}},k}\left|\frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p\,\partial_{x_1}\phi_{\boldsymbol{\theta}}^k(X_p)\right|\right]\\
&= M\,\mathbb{E}_{\mathcal{D}_{\sigma,X}}\left[\sup_{\boldsymbol{\theta}\in\Theta}\left|\frac{1}{N_{\rm in}}\sum_{p=1}^{N_{\rm in}} \sigma_p\,\partial_{x_1}\phi_{\boldsymbol{\theta}}^1(X_p)\right|\right] = M\,\hat{\mathfrak{R}}_{N_{\rm in}}(\mathcal{F}_{1,sub})\,.
\end{aligned}$$

Similarly, we have

$$\mathfrak{R}_{N_{\rm in}}(\mathcal{F}_2) \le M\,\hat{\mathfrak{R}}_{N_{\rm in}}(\mathcal{F}_{2,sub})\,, \qquad \mathfrak{R}_{N_b}(\mathcal{F}_3) \le M\,\hat{\mathfrak{R}}_{N_b}(\mathcal{F}_{3,sub})\,.$$

Combining above steps then concludes the proof. ■

Remark 10 *The above conclusion reveals an important fact: the Rademacher complexity of $\mathcal{F}_i$ is controlled by M and the complexity of $\mathcal{F}_{i,sub}$. This suggests that the network's overall complexity may not be affected by the number of sub-networks $\mathfrak{m}$, aiding us in managing the statistical error within the over-parameterized setting, where $\mathfrak{m}$ can grow arbitrarily large.*

B.3 Bounding the Rademacher Complexity Through Covering Number

We first provide the definition of the 'covering number', and the related Lemma 14, which is an effective tool that can further help us control $\hat{\mathfrak{R}}_N(\mathcal{F}_{i,sub})$. The proof of Lemma 14 can be found in Appendix B.9.

Definition 5 *An ϵ-cover of a set T in a metric space (S,τ) is a subset $T_c \subset S$ such that for each $t \in T$, there exists $t_c \in T_c$ such that $\tau(t,t_c) \leq \epsilon$. The ϵ-covering number of T, denoted as $\mathcal{C}(\epsilon,T,\tau)$ is defined to be the minimum cardinality among all ϵ-cover of T with respect to the metric τ.*

Lemma 14 *Let $\mathcal{F}$ be a class of functions mapping from Ω to $\mathbb{R}$ with $0 \in \mathcal{F}$, while for any $f \in \mathcal{F}$, we have $\|f\|_{L^\infty(\Omega)} \leq \mathcal{B}$. Then, it holds that*

$$\hat{\mathfrak{R}}_N(\mathcal{F}) \leq \inf_{0<\delta<\mathcal{B}} \left(4\delta + \frac{12}{\sqrt{N}} \int_\delta^{\mathcal{B}} \sqrt{\log 2\mathcal{C}(\epsilon,\mathcal{F},\|\cdot\|_{L^\infty})}\,\mathrm{d}\epsilon\right).$$

A Lipschitz parameterization allows us to translate a cover of the function space into a cover of the parameter space. The following result is direct.

Lemma 15 *Let $\mathcal{F}$ be a parameterized function class, $\mathcal{F} = \{f(\cdot;\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$. Let $\|\cdot\|_\Theta$ be a norm on Θ and let $\|\cdot\|_\mathcal{F}$ be a norm on $\mathcal{F}$. Suppose that the mapping $\boldsymbol{\theta} \mapsto f(\cdot;\boldsymbol{\theta})$ is λ-Lipschitz, that is,*

$$\left\|f(\cdot;\boldsymbol{\theta}) - f\left(\cdot;\tilde{\boldsymbol{\theta}}\right)\right\|_\mathcal{F} \leq \lambda \left\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\right\|_\Theta,$$

then for any $\epsilon > 0$, it holds that

$$\mathcal{C}(\epsilon,\mathcal{F},\|\cdot\|_\mathcal{F}) \leq \mathcal{C}(\epsilon\lambda^{-1},\Theta,\|\cdot\|_\Theta).$$

In Euclidean space, we can establish an upper bound of covering number for a bounded set easily. The following result is also direct.

Lemma 16 *Suppose that $T \subset \mathbb{R}^d$ and $\|t\|_2 \leq B$ for $t \in T$, it holds that*

$$\mathcal{C}(\epsilon,T,\|\cdot\|_2) \leq \left(2B\sqrt{d}\epsilon^{-1}\right)^d.$$

Based on above results, we present the following lemma, providing an upper bound for $\hat{\mathfrak{R}}_N(\mathcal{F}_{i,sub})$ in terms of the covering number of $\mathcal{F}_{i,sub}$.

Lemma 17 *Let $N_{\text{in}} = N_b = N_s$. For $i = 1,2,3$, it holds*

$$\hat{\mathfrak{R}}_{N_s}(\mathcal{F}_{i,sub}) \leq C(W,L)B_{\boldsymbol{\theta}}^L N_s^{-1/2}\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)},$$

Proof By Lemma 14, for $i = 1,2,3$, it holds that

$$\hat{\mathfrak{R}}_N(\mathcal{F}_{i,sub}) \leq \inf_{0<\delta<B_i} \left(4\delta + \frac{12}{\sqrt{N}} \int_\delta^{B_i} \sqrt{\log 2\mathcal{C}(\epsilon,\mathcal{F}_{i,sub},\|\cdot\|_{L^\infty})}\,\mathrm{d}\epsilon\right).$$

Combining Lemmas 9 and 15, we have

$$\mathcal{C}(\epsilon,\mathcal{F}_{i,sub},\|\cdot\|_{L^\infty}) \leq \mathcal{C}(\epsilon L_i^{-1},\Theta,\|\cdot\|_2),$$

and each of them can be bounded through Lemma 16:

$$\mathcal{C}(\epsilon, \mathcal{F}_{i,sub}, \|\cdot\|_{L^\infty}) \leq \left(2B_{\boldsymbol{\theta}} L_i \sqrt{W(W+1)L}\epsilon^{-1}\right)^{W(W+1)L}.$$

The above B_i and L_i denote the boundedness and smoothness indices of $\mathcal{F}_{i,sub}$ with respect to model parameters, as shown in Lemma 9. For instance, to upper bound $\hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{1,sub})$, we substitute B_1 and L_1 from Lemma 9 and obtain

$$\begin{aligned}
\hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{1,sub}) &\leq \inf_{0<\delta<B_1}\left(4\delta + \frac{12}{\sqrt{N_{\text{in}}}}\int_\delta^{B_1}\sqrt{\log 2\mathcal{C}(\epsilon, \mathcal{F}_{1,sub}, \|\cdot\|_{L^\infty})}\,\mathrm{d}\epsilon\right)\\
&\leq \inf_{0<\delta<B_1}\left\{4\delta + \frac{12}{\sqrt{N_{\text{in}}}}\int_\delta^{B_1}\sqrt{W(W+1)L}\log^{1/2}\left[4L(L+1)W^{2L}B_{\boldsymbol{\theta}}^{2L+1}\epsilon^{-1}\right]\mathrm{d}\epsilon\right\}\\
&\leq \inf_{0<\delta<B_1}\left\{4\delta + 12W\sqrt{L}B_1 N_{\text{in}}^{-1/2}\log^{1/2}\left[4L(L+1)W^{2L}B_{\boldsymbol{\theta}}^{2L+1}\delta^{-1}\right]\right\}.
\end{aligned}$$

Choosing $\delta = N_{\text{in}}^{-1/2} < B_1/2$, we have

$$\hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{1,sub}) \leq C(W,L)B_{\boldsymbol{\theta}}^L N_{\text{in}}^{-1/2}\sqrt{\log(B_{\boldsymbol{\theta}}WLN_{\text{in}})}\,.$$

$\hat{\mathfrak{R}}_{N_{\text{in}}}(\mathcal{F}_{2,sub})$ and $\hat{\mathfrak{R}}_{N_b}(\mathcal{F}_{3,sub})$ can also be bounded in a similar way. ■

B.4 Complete Statistical Error Bound

Combining previous analysis and with the help of the following inequality, we could finally obtain high probability control over $\mathcal{E}_{sta}$. Proof of Lemma 18 is referred to McDiarmid et al. (1989).

Lemma 18 *Let g be a function from $\Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ to $\mathbb{R}$. Suppose that function g satisfies the bounded differences property, that is, there exists constants $c_1, \ldots, c_n > 0$ such that for any $x_1 \in \Omega_1, \ldots, x_n \in \Omega_n$*

$$\sup_{\tilde{x}_i \in \Omega_i} |g(x_1, \cdots, \tilde{x}_i, \cdots, x_n) - g(x_1, \cdots, x_i, \cdots, x_n)| \leq c_i\,, \quad i = 1, \ldots, n\,.$$

Let $\{X_i\}_{i=1}^n$ be independent variables, where $X_i \in \Omega_i$, then for any $\tau > 0$, we have

$$|g(X_1, \cdots, X_n) - \mathbb{E}[g(X_1, \cdots, X_n)]| \leq \tau$$

with probability at least $1 - 2\exp[-2\tau^2(\sum_{i=1}^n c_i^2)^{-1}]$.

Theorem 8 (**Theorem 5** in the main text) *Let $\mathcal{PNN} = \mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$. Let $N_{\text{in}} = N_b = N_s$ in the Monte Carlo sampling. Let $0 < \xi < 1$. Then, with probability at least $1 - \xi$, it holds that*

$$\begin{aligned}
\mathcal{E}_{sta} &= \sup_{u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}} \left|\mathcal{L}(u_{\mathfrak{m},\boldsymbol{\theta}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})\right|\\
&\leq C(\Omega, B_0, d, W, L)\cdot M^2 B_{\boldsymbol{\theta}}^{2L} N_s^{-1/2}\left(\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)} + \sqrt{\log \xi^{-1}}\right),
\end{aligned}$$

where $C(\Omega, B_0, d, W, L)$ is a universal constant which only depends on Ω, B_0, d, W and L.

Proof By Lemmas 10 and 13, we have

$$\begin{aligned}\mathbb{E}[\mathcal{E}_{sta}] \leq\ & 4d\,|\Omega|\,W^{L-1}\,M^2\,B_{\boldsymbol{\theta}}^{L}\cdot\hat{\mathfrak{R}}_{N_{\mathrm{in}}}(\mathcal{F}_{1,sub})\\ &+4B_0\,|\Omega|\,\big[M(W+1)B_{\boldsymbol{\theta}}+1\big]\,M\cdot\hat{\mathfrak{R}}_{N_{\mathrm{in}}}(\mathcal{F}_{2,sub})+2B_0|\partial\Omega|M\cdot\hat{\mathfrak{R}}_{N_b}(\mathcal{F}_{3,sub})\,.\end{aligned}$$

By Lemma 17, for $i=1,2,3$, it holds that

$$\hat{\mathfrak{R}}_{N_s}(\mathcal{F}_{i,sub})\leq C(W,L)B_{\boldsymbol{\theta}}^{L}N_s^{-1/2}\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)}\,.$$

Then, we get

$$\mathbb{E}[\mathcal{E}_{sta}]\leq C_1(\Omega,B_0,d,W,L)M^2B_{\boldsymbol{\theta}}^{2L}N_s^{-1/2}\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)}\,. \tag{37}$$

Now, we denote by

$$\gamma(X_1,\ldots,X_{N_s},Y_1,\ldots,Y_{N_s}):=\sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}}|\mathcal{L}(u_{\mathfrak{m},\boldsymbol{\theta}})-\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})|=\mathcal{E}_{sta}\,.$$

Examining the difference of $\gamma(X_1,\ldots,X_{N_s},Y_1,\ldots,Y_{N_s})$, we have

$$\begin{aligned}&|\gamma(X_1,\ldots,X_i,\ldots,Y_{N_s})-\gamma(X_1,\ldots,X_i',\ldots,Y_{N_s})|\\ &\leq\frac{|\Omega|}{N_s}\sup_{u\in\mathcal{PNN}}\bigg|\frac{\|\nabla u(X_i)\|_2^2-\|\nabla u(X_i')\|_2^2}{2}+\frac{w(X_i)u^2(X_i)-w(X_i')u^2(X_i')}{2}\\ &\qquad\qquad+u(X_i')h(X_i')-u(X_i)h(X_i)\bigg|\\ &\leq 4|\Omega|N_s^{-1}dM^2\,(B_0+1)W^{2L-2}B_{\boldsymbol{\theta}}^{2L}\,,\end{aligned}$$

where we have used the boundedness properties outlined in Lemma 9. We also have

$$|\gamma(X_1,\ldots,Y_j,\ldots,Y_{N_s})-\gamma(X_1,\ldots,Y_j',\ldots,Y_{N_s})|\leq 2|\partial\Omega|N_s^{-1}B_0M(W+1)B_{\boldsymbol{\theta}}\,.$$

Then by Lemma 18 and Equation 37, it holds that

$$\mathcal{E}_{sta}\leq\mathbb{E}[\mathcal{E}_{sta}]+\tau\leq C_1(\Omega,B_0,d,W,L)M^2B_{\boldsymbol{\theta}}^{2L}N_s^{-1/2}\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)}+\tau$$

with probability as least

$$1-2\exp\bigg\{-\frac{N_s\tau^2}{8d^2(|\partial\Omega|^2+|\Omega|^2)(B_0+1)^2W^{4L}B_{\boldsymbol{\theta}}^{4L}M^4}\bigg\}\,.$$

This implies that with probability at least $1-\xi$, we have

$$\begin{aligned}&\sup_{u_{\mathfrak{m},\boldsymbol{\theta}}\in\mathcal{PNN}}\big|\mathcal{L}(u_{\mathfrak{m},\boldsymbol{\theta}})-\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})\big|\\ &\qquad\leq C_2(\Omega,B_0,d,W,L)M^2B_{\boldsymbol{\theta}}^{2L}N_s^{-1/2}\big(\sqrt{\log(B_{\boldsymbol{\theta}}WLN_s)}+\sqrt{\log\xi^{-1}}\big)\,,\end{aligned}$$

which concludes the proof. ■

B.5 Proof of Lemma 7

By definition in Section 2.2, for any $\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{NN}(W, L, B_{\boldsymbol{\theta}})$, we have

$$\big|\phi_{\boldsymbol{\theta}}(\boldsymbol{x})\big| \le \big(N_{L-1}+1\big)B_{\boldsymbol{\theta}} \le \big(W+1\big)B_{\boldsymbol{\theta}}\,.$$

Use $\phi_i^{(\ell)}$ to denote the i-th output of the ℓ-th layer. Note that $\rho = \tanh$ is $1-$Lipshcitz. For $\ell = 2, \ldots, L$, it holds that

$$\begin{aligned}
\big|\phi_i^{(\ell)} - \tilde{\phi}_i^{(\ell)}\big| &= \bigg|\rho\Big(\sum_{j=1}^{N_{\ell-1}} a_{ij}^{(\ell-1)}\phi_j^{(\ell-1)} + b_i^{(\ell-1)}\Big) - \rho\Big(\sum_{j=1}^{N_{\ell-1}} \tilde{a}_{ij}^{(\ell-1)}\tilde{\phi}_j^{(\ell-1)} + \tilde{b}_i^{(\ell-1)}\Big)\bigg| \\
&\le \bigg|\sum_{j=1}^{N_{\ell-1}} a_{ij}^{(\ell-1)}\phi_j^{(\ell-1)} - \sum_{j=1}^{N_{\ell-1}} \tilde{a}_{ij}^{(\ell-1)}\tilde{\phi}_j^{(\ell-1)} + b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\bigg| \\
&\le \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)}\big|\big|\phi_j^{(\ell-1)} - \tilde{\phi}_j^{(\ell-1)}\big| + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big|\big|\tilde{\phi}_j^{(\ell-1)}\big| + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\phi_j^{(\ell-1)} - \tilde{\phi}_j^{(\ell-1)}\big| + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big| + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big|\,.
\end{aligned}$$

For $\ell = 1$, we have

$$\big|\phi_i^{(1)} - \tilde{\phi}_i^{(1)}\big| \le \sum_{j=1}^{N_0} \big|a_{ij}^{(0)} - \tilde{a}_{ij}^{(0)}\big| + \big|b_i^{(0)} - \tilde{b}_i^{(0)}\big| = \sum_{j=1}^{\mathfrak{n}_1} \big|\theta_j - \tilde{\theta}_j\big|\,.$$

For $\ell = 2$, we have

$$\begin{aligned}
\big|\phi_i^{(2)} - \tilde{\phi}_i^{(2)}\big| &\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_1} \big|\phi_j^{(1)} - \tilde{\phi}_j^{(1)}\big| + \sum_{j=1}^{N_1} \big|a_{ij}^{(1)} - \tilde{a}_{ij}^{(1)}\big| + \big|b_i^{(1)} - \tilde{b}_i^{(1)}\big| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_1}\sum_{k=1}^{\mathfrak{n}_1} \big|\theta_k - \tilde{\theta}_k\big| + \sum_{j=1}^{N_1} \big|a_{ij}^{(1)} - \tilde{a}_{ij}^{(1)}\big| + \big|b_i^{(1)} - \tilde{b}_i^{(1)}\big| \le N_1 B_{\boldsymbol{\theta}} \sum_{j=1}^{\mathfrak{n}_2} \big|\theta_j - \tilde{\theta}_j\big|\,.
\end{aligned}$$

Assuming that for $\ell \ge 2$, it holds that

$$\big|\phi_i^{(\ell)} - \tilde{\phi}_i^{(\ell)}\big| \le \Big(\prod_{i=1}^{\ell-1} N_i\Big) B_{\boldsymbol{\theta}}^{\ell-1} \sum_{j=1}^{\mathfrak{n}_\ell} \big|\theta_j - \tilde{\theta}_j\big|\,,$$

Then, we have

$$\begin{aligned}
\big|\phi_i^{(\ell+1)} - \tilde{\phi}_i^{(\ell+1)}\big| &\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_\ell} \big|\phi_j^{(\ell)} - \tilde{\phi}_j^{(\ell)}\big| + \sum_{j=1}^{N_\ell} \big|a_{ij}^{(\ell)} - \tilde{a}_{ij}^{(\ell)}\big| + \big|b_i^{(\ell)} - \tilde{b}_i^{(\ell)}\big| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_\ell} \Big(\prod_{i=1}^{\ell-1} N_i\Big) B_{\boldsymbol{\theta}}^{\ell-1} \sum_{k=1}^{\mathfrak{n}_\ell} \big|\theta_k - \tilde{\theta}_k\big| + \sum_{j=1}^{N_\ell} \big|a_{ij}^{(\ell)} - \tilde{a}_{ij}^{(\ell)}\big| + \big|b_i^{(\ell)} - \tilde{b}_i^{(\ell)}\big|
\end{aligned}$$

$$\leq \Big(\prod_{i=1}^{\ell} N_i\Big) B_{\boldsymbol{\theta}}^{\ell} \sum_{j=1}^{\mathfrak{n}_{\ell+1}} \big|\theta_j - \tilde{\theta}_j\big| .$$

Hence, by induction, we could conclude that

$$\begin{aligned}
|\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) - \phi_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x})| &\leq \Big(\prod_{i=1}^{L-1} N_i\Big) B_{\boldsymbol{\theta}}^{L-1} \sum_{j=1}^{\mathfrak{n}_L} \big|\theta_j - \tilde{\theta}_j\big| \\
&\leq \sqrt{\mathfrak{n}_L} B_{\boldsymbol{\theta}}^{L-1} \Big(\prod_{i=1}^{L-1} N_i\Big) \big\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\big\|_2 \leq 2W^L \sqrt{L} B_{\boldsymbol{\theta}}^{L-1} \big\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\big\|_2 .
\end{aligned}$$

B.6 Proof of Lemma 8

Use $\phi_i^{(\ell)}$ to denote the i-th output of the ℓ-th layer. For $\rho = \tanh$, note that ρ and ρ' are both 1−Lipshcitz. For $\ell = 1, 2, \cdots, L-1$, it holds that

$$\begin{aligned}
\big|\partial_{x_m} \phi_i^{(\ell)}\big| &= \Big| \sum_{j=1}^{N_{\ell-1}} a_{ij}^{(\ell-1)} \partial_{x_m} \phi_j^{(\ell-1)} \rho'\Big(\sum_{j=1}^{N_{\ell-1}} a_{ij}^{(\ell-1)} \phi_j^{(\ell-1)} + b_i^{(\ell-1)} \Big) \Big| \leq B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m} \phi_j^{(\ell-1)}\big| \\
&\leq (B_{\boldsymbol{\theta}})^2 \sum_{k=1}^{N_{\ell-1}} \sum_{j=1}^{N_{\ell-2}} \big|\partial_{x_m} \phi_j^{(\ell-2)}\big| = N_{\ell-1} (B_{\boldsymbol{\theta}})^2 \sum_{j=1}^{N_{\ell-2}} \big|\partial_{x_m} \phi_j^{(\ell-2)}\big| \\
&\leq \cdots \leq \Big(\prod_{i=2}^{\ell-1} N_i\Big) (B_{\boldsymbol{\theta}})^{\ell-1} \sum_{j=1}^{N_1} \big|\partial_{x_m} \phi_j^{(1)}\big| \\
&\leq \Big(\prod_{i=2}^{\ell-1} N_i\Big) (B_{\boldsymbol{\theta}})^{\ell-1} \sum_{j=1}^{N_1} B_{\boldsymbol{\theta}} = \Big(\prod_{i=1}^{\ell-1} N_i\Big) (B_{\boldsymbol{\theta}})^{\ell} \leq W^{\ell-1} B_{\boldsymbol{\theta}}^{\ell} .
\end{aligned}$$

The bound for $|\partial_{x_m} \phi_{\boldsymbol{\theta}}(\boldsymbol{x})|$ can be derived similarly. For $\ell = 1$, we have

$$\begin{aligned}
\big|\partial_{x_m} \phi_i^{(1)} - \partial_{x_m} \tilde{\phi}_i^{(1)}\big| &= \Big| a_{im}^{(0)} \rho'\Big(\sum_{j=1}^{N_0} a_{ij}^{(0)} x_j + b_i^{(0)} \Big) - \tilde{a}_{im}^{(0)} \rho'\Big(\sum_{j=1}^{N_0} \tilde{a}_{ij}^{(0)} x_j + \tilde{b}_i^{(0)} \Big) \Big| \\
&\leq \big|a_{im}^{(0)} - \tilde{a}_{im}^{(0)}\big| \Big| \rho'\Big(\sum_{j=1}^{N_0} a_{ij}^{(0)} x_j + b_i^{(0)} \Big) \Big| \\
&\qquad + \big|\tilde{a}_{im}^{(0)}\big| \Big| \rho'\Big(\sum_{j=1}^{N_0} a_{ij}^{(0)} x_j + b_i^{(0)} \Big) - \rho'\Big(\sum_{j=1}^{N_0} \tilde{a}_{ij}^{(0)} x_j + \tilde{b}_i^{(0)} \Big) \Big| \\
&\leq \big|a_{im}^{(0)} - \tilde{a}_{im}^{(0)}\big| + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_0} \big|a_{ij}^{(0)} - \tilde{a}_{ij}^{(0)}\big| + B_{\boldsymbol{\theta}} \big|b_i^{(0)} - \tilde{b}_i^{(0)}\big| \leq 2B_{\boldsymbol{\theta}} \sum_{k=1}^{\mathfrak{n}_1} \big|\theta_k - \tilde{\theta}_k\big| .
\end{aligned}$$

For $\ell \geq 2$, we establish the recurrence relation:

$$\big|\partial_{x_m} \phi_i^{(\ell)} - \partial_{x_m} \tilde{\phi}_i^{(\ell)}\big|$$

$$
\begin{aligned}
&\le \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)}\big|\big|\partial_{x_m}\phi_j^{(\ell-1)}\big|\bigg|\rho'\Big(\sum_{j=1}^{N_{\ell-1}} a_{ij}^{(\ell-1)}\phi_j^{(\ell-1)} + b_i^{(\ell-1)}\Big) - \rho'\Big(\sum_{j=1}^{N_{\ell-1}} \tilde{a}_{ij}^{(\ell-1)}\tilde{\phi}_j^{(\ell-1)} + \tilde{b}_i^{(\ell-1)}\Big)\bigg| \\
&\quad + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)}\partial_{x_m}\phi_j^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big|\bigg|\rho'\Big(\sum_{j=1}^{N_{\ell-1}} \tilde{a}_{ij}^{(\ell-1)}\tilde{\phi}_j^{(\ell-1)} + \tilde{b}_i^{(\ell-1)}\Big)\bigg| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)}\big|\Big(\sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)}\phi_j^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\tilde{\phi}_j^{(\ell-1)}\big| + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big|\Big) \\
&\quad + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)}\partial_{x_m}\phi_j^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)}\big|\Big(\sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big| + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\phi_j^{(\ell-1)} - \tilde{\phi}_j^{(\ell-1)}\big| + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big|\Big) \\
&\quad + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)} - \partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big|\big|\partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)}\big|\Big(\sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big| + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\phi_j^{(\ell-1)} - \tilde{\phi}_j^{(\ell-1)}\big| + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big|\Big) \\
&\quad + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)} - \partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big|\big|\partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| \\
&\le B_{\boldsymbol{\theta}}\Big(\prod_{i=1}^{\ell-1} N_i\Big)B_{\boldsymbol{\theta}}^{\ell}\Big[\sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big| + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \Big(\prod_{i=1}^{\ell-2} N_i\Big)B_{\boldsymbol{\theta}}^{\ell-2} \sum_{k=1}^{\mathfrak{n}_{\ell-1}} \big|\theta_k - \tilde{\theta}_k\big| \\
&\quad + \big|b_i^{(\ell-1)} - \tilde{b}_i^{(\ell-1)}\big|\Big] + B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\big(\phi_j^{(\ell-1)} - \tilde{\phi}_j^{(\ell-1)}\big)\big| + \sum_{j=1}^{N_{\ell-1}} \big|a_{ij}^{(\ell-1)} - \tilde{a}_{ij}^{(\ell-1)}\big|\Big(\prod_{i=1}^{\ell-2} N_i\Big)B_{\boldsymbol{\theta}}^{\ell-1} \\
&\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_{\ell-1}} \big|\partial_{x_m}\phi_j^{(\ell-1)} - \partial_{x_m}\tilde{\phi}_j^{(\ell-1)}\big| + B_{\boldsymbol{\theta}}^{2\ell}\Big(\prod_{i=1}^{\ell-1} N_i\Big)^2 \sum_{k=1}^{\mathfrak{n}_\ell} \big|\theta_k - \tilde{\theta}_k\big|.
\end{aligned}
$$

When $\ell = 2$, we have

$$
\begin{aligned}
\big|\partial_{x_m}\phi_i^{(2)} - \partial_{x_m}\tilde{\phi}_i^{(2)}\big| &\le B_{\boldsymbol{\theta}} \sum_{j=1}^{N_1} \big|\partial_{x_m}\phi_j^{(1)} - \partial_{x_m}\tilde{\phi}_j^{(1)}\big| + B_{\boldsymbol{\theta}}^4 N_1^2 \sum_{k=1}^{\mathfrak{n}_2} \big|\theta_k - \tilde{\theta}_k\big| \\
&\le 2B_{\boldsymbol{\theta}}^2 N_1 \sum_{k=1}^{\mathfrak{n}_1} \big|\theta_k - \tilde{\theta}_k\big| + B_{\boldsymbol{\theta}}^4 N_1^2 \sum_{k=1}^{\mathfrak{n}_2} \big|\theta_k - \tilde{\theta}_k\big| \le 3B_{\boldsymbol{\theta}}^4 N_1^2 \sum_{k=1}^{\mathfrak{n}_2} \big|\theta_k - \tilde{\theta}_k\big|.
\end{aligned}
$$

Assuming that for $\ell \ge 2$, it holds that

$$
\big|\partial_{x_m}\phi_i^{(\ell)} - \partial_{x_m}\tilde{\phi}_i^{(\ell)}\big| \le (\ell+1)B_{\boldsymbol{\theta}}^{2\ell}\Big(\prod_{i=1}^{\ell-1} N_i\Big)^2 \sum_{k=1}^{\mathfrak{n}_\ell} \big|\theta_k - \tilde{\theta}_k\big|.
$$

Then, we have

$$\begin{aligned}
&\left|\partial_{x_m}\phi_i^{(\ell+1)} - \partial_{x_m}\tilde{\phi}_i^{(\ell+1)}\right| \\
&\quad\le B_{\boldsymbol{\theta}}\sum_{j=1}^{N_\ell}\left|\partial_{x_m}\phi_j^{(\ell)} - \partial_{x_m}\tilde{\phi}_j^{(\ell)}\right| + B_{\boldsymbol{\theta}}^{2\ell+2}\bigg(\prod_{i=1}^{\ell}N_i\bigg)^2\sum_{k=1}^{\mathfrak{n}_{\ell+1}}\left|\theta_k-\tilde{\theta}_k\right| \\
&\quad\le B_{\boldsymbol{\theta}}\sum_{j=1}^{N_\ell}(\ell+1)B_{\boldsymbol{\theta}}^{2\ell}\bigg(\prod_{i=1}^{\ell-1}N_i\bigg)^2\sum_{k=1}^{\mathfrak{n}_\ell}\left|\theta_k-\tilde{\theta}_k\right| + B_{\boldsymbol{\theta}}^{2\ell+2}\bigg(\prod_{i=1}^{\ell}N_i\bigg)^2\sum_{k=1}^{\mathfrak{n}_{\ell+1}}\left|\theta_k-\tilde{\theta}_k\right| \\
&\quad\le (\ell+2)B_{\boldsymbol{\theta}}^{2\ell+2}\bigg(\prod_{i=1}^{\ell}N_i\bigg)^2\sum_{k=1}^{\mathfrak{n}_{\ell+1}}\left|\theta_k-\tilde{\theta}_k\right|.
\end{aligned}$$

Hence, by induction, we could conclude that

$$\begin{aligned}
\left|\partial_{x_m}\phi_{\boldsymbol{\theta}} - \partial_{x_m}\phi_{\tilde{\boldsymbol{\theta}}}\right| = \left|\partial_{x_m}\phi - \partial_{x_m}\tilde{\phi}\right| &\le (L+1)B_{\boldsymbol{\theta}}^{2L}\bigg(\prod_{i=1}^{L-1}N_i\bigg)^2\sum_{k=1}^{\mathfrak{n}_L}\left|\theta_k-\tilde{\theta}_k\right| \\
&\le \sqrt{\mathfrak{n}_L}(L+1)B_{\boldsymbol{\theta}}^{2L}\bigg(\prod_{i=1}^{L-1}N_i\bigg)^2\left\|\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}\right\|_2 \le 2W^{2L-1}\sqrt{L}(L+1)B_{\boldsymbol{\theta}}^{2L}\left\|\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}\right\|_2.
\end{aligned}$$

B.7 Proof of Lemma 11

Denote by $\mathbb{E}_{X,\sigma} := \mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^N}$ and $\mathbb{E}_{X,\bar{\sigma}} := \mathbb{E}_{\{X_k\}_{k=1}^N,\{\sigma_k\}_{k=2}^N}$. Recursively, we have

$$\begin{aligned}
\mathfrak{R}_N(a\cdot\mathcal{F}) &= \frac{1}{N}\mathbb{E}_{X,\sigma}\bigg[\sup_{f\in\mathcal{F}}\sum_{k=1}^{N}\sigma_k a(X_k)f(X_k)\bigg] \\
&= \frac{1}{2N}\mathbb{E}_{X,\bar{\sigma}}\bigg[\sup_{f\in\mathcal{F}}\bigg[a(X_1)f(X_1)+\sum_{k=2}^{N}\sigma_k a(X_k)f(X_k)\bigg]\bigg] \\
&\qquad + \frac{1}{2N}\mathbb{E}_{X,\bar{\sigma}}\bigg[\sup_{f\in\mathcal{F}}\bigg[-a(X_1)f(X_1)+\sum_{k=2}^{N}\sigma_k a(X_k)f(X_k)\bigg]\bigg] \\
&= \frac{1}{2N}\mathbb{E}_{X,\bar{\sigma}}\bigg[\sup_{f,f'\in\mathcal{F}}\bigg[a(X_1)(f-f')(X_k)+\sum_{k=2}^{N}\sigma_k a(X_k)(f+f')(X_k)\bigg]\bigg] \\
&\le \frac{1}{2N}\mathbb{E}_{X,\bar{\sigma}}\bigg[\sup_{f,f'\in\mathcal{F}}\bigg[\mathcal{B}|(f-f')(X_k)|+\sum_{k=2}^{N}\sigma_k a(X_k)(f+f')(X_k)\bigg]\bigg] \\
&= \frac{1}{2N}\mathbb{E}_{X,\bar{\sigma}}\bigg[\sup_{f,f'\in\mathcal{F}}\bigg[\mathcal{B}(f-f')(X_k)+\sum_{k=2}^{N}\sigma_k a(X_k)(f+f')(X_k)\bigg]\bigg] \\
&= \frac{1}{N}\mathbb{E}_{X,\sigma}\bigg[\sup_{f\in\mathcal{F}}\bigg[\sigma_1\mathcal{B}f(X_1)+\sum_{k=2}^{N}\sigma_k a(X_k)f(X_k)\bigg]\bigg] \\
&\le \cdots \le \frac{\mathcal{B}}{N}\mathbb{E}_{X,\sigma}\bigg[\sup_{f\in\mathcal{F}}\sum_{k=1}^{N}\sigma_k f(X_k)\bigg] = \mathcal{B}\mathfrak{R}_N(\mathcal{F}).
\end{aligned}$$

B.8 Proof of Lemma 12

For fixed $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, denote by $h_{N-1}(f) := \sum_{i=1}^{N-1} \sigma_i(\Phi \circ f)(\boldsymbol{x}_i)$. We have

$$\mathbb{E}_{\{\sigma_k\}_{k=1}^N}\Big[\sup_{f\in\mathcal{F}}\sum_{i=1}^{N}\sigma_i(\Phi\circ f)(\boldsymbol{x}_i)\Big] = \mathbb{E}_{\{\sigma_k\}_{k=1}^{N-1}}\mathbb{E}_{\sigma_N}\Big[\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+\sigma_N(\Phi\circ f)(\boldsymbol{x}_N)\big]\Big],$$

For any $\epsilon > 0$, there exist $f_1, f_2 \in \mathcal{F}$ such that

$$h_{N-1}(f_1) + (\Phi\circ f_1)(\boldsymbol{x}_N) \geq (1-\epsilon)\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+(\Phi\circ f)(\boldsymbol{x}_N)\big],$$
$$h_{N-1}(f_2) - (\Phi\circ f_2)(\boldsymbol{x}_N) \geq (1-\epsilon)\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)-(\Phi\circ f)(\boldsymbol{x}_N)\big].$$

Thus, it holds that

$$\begin{aligned}
&(1-\epsilon)\,\mathbb{E}_{\sigma_N}\Big[\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+\sigma_N(\Phi\circ f)(\boldsymbol{x}_N)\big]\Big]\\
&= \frac{1-\epsilon}{2}\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+(\Phi\circ f)(\boldsymbol{x}_N)\big] + \frac{1-\epsilon}{2}\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)-(\Phi\circ f)(\boldsymbol{x}_N)\big]\\
&\leq \frac{1}{2}\big[h_{N-1}(f_1)+(\Phi\circ f_1)(\boldsymbol{x}_N)\big] + \frac{1}{2}\big[h_{N-1}(f_2)-(\Phi\circ f_2)(\boldsymbol{x}_N)\big].
\end{aligned}$$

Let $s = \operatorname{sgn}(f_1(\boldsymbol{x}_N) - f_2(\boldsymbol{x}_N))$. Since Φ is λ-Lipschitz, we have

$$\begin{aligned}
&(1-\epsilon)\mathbb{E}_{\sigma_N}\Big[\sup_{f\in\mathcal{F}} h_{N-1}(f)+\sigma_N(\Phi\circ f)(\boldsymbol{x}_N)\Big]\\
&\quad\leq \frac{1}{2}\big\{h_{N-1}(f_1)+h_{N-1}(f_2)+s\lambda\big[f_1(\boldsymbol{x}_N)-f_2(\boldsymbol{x}_N)\big]\big\}\\
&\quad= \frac{1}{2}\big[h_{N-1}(f_1)+s\lambda f_1(\boldsymbol{x}_N)\big] + \frac{1}{2}\big[h_{N-1}(f_2)-s\lambda f_2(\boldsymbol{x}_N)\big]\\
&\quad\leq \frac{1}{2}\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+s\lambda f(\boldsymbol{x}_N)\big] + \frac{1}{2}\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)-s\lambda f(\boldsymbol{x}_N)\big]\\
&\quad= \mathbb{E}_{\sigma_N}\Big[\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+\sigma_N\lambda f(\boldsymbol{x}_N)\big]\Big],
\end{aligned}$$

Note that $\epsilon > 0$ is arbitrary. It holds that

$$\mathbb{E}_{\{\sigma_k\}_{k=1}^N}\Big[\sup_{f\in\mathcal{F}}\sum_{i=1}^{N}\sigma_i(\Phi\circ f)(\boldsymbol{x}_i)\Big] = \mathbb{E}_{\{\sigma_k\}_{k=1}^N}\Big[\sup_{f\in\mathcal{F}}\big[h_{N-1}(f)+\sigma_N\lambda f(\boldsymbol{x}_N)\big]\Big].$$

This technique is applied iteratively for all other σ_i $(i \neq N)$. Then, the lemma is proved by generalizing from fixed $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ to random $(X_1, \ldots, X_N)$, and subsequently taking expectation over $\{X_i\}_{i=1}^N$.

B.9 Proof of Lemma 14

Set $\epsilon_k = 2^{-k+1}\mathcal{B}$. Denote by $\mathcal{F}_k$ such that $\mathcal{F}_k$ is an ϵ_k-cover of $\mathcal{F}$ and $|\mathcal{F}_k| = \mathcal{C}(\epsilon_k, \mathcal{F}, \|\cdot\|_{L^\infty})$. For any $f \in \mathcal{F}$, there exists $f_k \in \mathcal{F}_k$, such that $\|f - f_k\|_{L^\infty} \leq \epsilon_k$. Let $K \in \mathbb{N}^+$. We have

$$\hat{\mathfrak{R}}_N(\mathcal{F}) = \mathbb{E}\Big[\sup_{f\in\mathcal{F}}\frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i f(X_i)\Big|\Big]$$

$$= \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i\Big[(f-f_K)(X_i)+\sum_{j=1}^{K-1}(f_{j+1}-f_j)(X_i)+f_1(X_i)\Big]\Big|\Big]$$
$$\le \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i\big[(f-f_K)(X_i)\big]\Big|\Big] + \mathbb{E}\Big[\sup_{f_1\in\mathcal{F}_1} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i f_1(X_i)\Big|\Big]$$
$$+ \sum_{j=1}^{K-1}\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i\big[(f_{j+1}-f_j)(X_i)\big]\Big|\Big].$$

We choose $\mathcal{F}_1=\{0\}$ to eliminate the second term above. For the first term, it holds that

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i\big[f(X_i)-f_K(X_i)\big]\Big|\Big] \le \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N}|\sigma_i|\|f-f_K\|_{L^\infty}\Big] \le \epsilon_K\,.$$

To handle the second term, for any fixed samples $\{X_i\}_{i=1}^N$ and $1\le j<K$, we define

$$V_j := \big\{(f_{j+1}(X_1)-f_j(X_1),\ldots,f_{j+1}(X_N)-f_j(X_N))\in\mathbb{R}^N : f\in\mathcal{F}\big\}\,.$$

Then, for any $\boldsymbol{v}^j=(v_1^j,\ldots,v_N^j)\in V_j$, it holds that

$$\|\boldsymbol{v}^j\|_2 = \Big(\sum_{i=1}^{N}|f_{j+1}(X_i)-f_j(X_i)|^2\Big)^{1/2} \le \sqrt{N}\|f_{j+1}-f_j\|_{L^\infty}$$
$$\le \sqrt{N}\|f_{j+1}-f\|_{L^\infty}+\sqrt{N}\|f_j-f\|_{L^\infty} = \sqrt{N}\epsilon_{j+1}+\sqrt{N}\epsilon_j = 3\sqrt{N}\epsilon_{j+1}\,.$$

Further, we present the following lemma, with proof provided in Appendix B.10.

Lemma 19 *Let $A\subseteq\mathbb{R}^m$ be finite, $\boldsymbol{x}=(x_1,\ldots,x_m)$ and $r=\max_{\boldsymbol{x}\in A}\|\boldsymbol{x}\|_2$. Then, we have*

$$\mathbb{E}\Big[\frac{1}{m}\sup_{\boldsymbol{x}\in A}\Big|\sum_{i=1}^{m}\sigma_i x_i\Big|\Big] \le \frac{r\sqrt{2\log(2|A|)}}{m}\,,$$

where $\{\sigma_i\}_{i=1}^m$ are independent Rademacher variables.

Applying Lemma 19 and take partial expectation with respect to $\{\sigma_i\}_{i=1}^N$, we have

$$\sum_{j=1}^{K-1}\mathbb{E}_{\{\sigma_i\}_{i=1}^N}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i\big[f_{j+1}(X_i)-f_j(X_i)\big]\Big|\Big]$$
$$= \sum_{j=1}^{K-1}\mathbb{E}_{\{\sigma_i\}_{i=1}^N}\Big[\sup_{\boldsymbol{v}^j\in V_j} \frac{1}{N}\Big|\sum_{i=1}^{N}\sigma_i v_i^j\Big|\Big] \le \sum_{j=1}^{K-1}\frac{3\epsilon_{j+1}}{\sqrt{N}}\sqrt{2\log(2|V_j|)}\,.$$

By the definition of V_j, we know that $|V_j|\le|\mathcal{F}_j||\mathcal{F}_{j+1}|\le|\mathcal{F}_{j+1}|^2$. Hence, it holds that

$$\sum_{j=1}^{K-1}\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N}\sigma_i\big[f_{j+1}(X_i)-f_j(X_i)\big]\Big] \le \sum_{j=1}^{K-1}\frac{6\epsilon_{j+1}}{\sqrt{N}}\sqrt{\log(2|\mathcal{F}_{j+1}|)}\,.$$

Finally, we obtain

$$\begin{aligned}\hat{\mathfrak{R}}_N(\mathcal{F}) &\le \epsilon_K + \sum_{j=1}^{K-1} \frac{6\epsilon_{j+1}}{\sqrt{N}} \sqrt{\log(2|\mathcal{F}_{j+1}|)} \\ &= \epsilon_K + \frac{12}{\sqrt{N}} \sum_{j=1}^{K} (\epsilon_j - \epsilon_{j+1}) \sqrt{\log 2\mathcal{C}(\epsilon_j, \mathcal{F}, \|\cdot\|_{L^\infty})} \\ &\le \epsilon_K + \frac{12}{\sqrt{N}} \int_{\epsilon_{K+1}}^{\mathcal{B}} \sqrt{\log 2\mathcal{C}(\epsilon, \mathcal{F}, \|\cdot\|_{L^\infty})}\, \mathrm{d}\epsilon \\ &\le \inf_{0<\delta<\mathcal{B}} \left[4\delta + \frac{12}{\sqrt{N}} \int_{\delta}^{\mathcal{B}} \sqrt{\log 2\mathcal{C}(\epsilon, \mathcal{F}, \|\cdot\|_{L^\infty})}\, \mathrm{d}\epsilon \right].\end{aligned}$$

where the last inequality holds because for $0 \le \delta \le \mathcal{B}$, we can choose K as the largest integer such that $\epsilon_{K+1} > \delta$. This choice implies that $\epsilon_K \le 4\epsilon_{K+2} \le 4\delta$.

B.10 Proof of Lemma 19

For any $t > 0$, using Jensen's inequality, rearranging terms, we obtain

$$\begin{aligned}\exp\left[t\mathbb{E}\Big(\sup_{x\in A} \Big| \sum_{i=1}^m \sigma_i x_i \Big| \Big) \right] &\le \mathbb{E}\left[\exp\left(t \sup_{x\in A} \Big| \sum_{i=1}^m \sigma_i x_i \Big| \right) \right] \\ &= \mathbb{E}\left[\sup_{x\in A} \exp\left(t \Big| \sum_{i=1}^m \sigma_i x_i \Big| \right) \right] \le \sum_{x\in A} \mathbb{E}\left[\exp\left(t \Big| \sum_{i=1}^m \sigma_i x_i \Big| \right) \right].\end{aligned}$$

Since $e^{|x|} \le e^x + e^{-x}$ and $e^x + e^{-x} \le 2e^{x^2/2}$, it holds that

$$\begin{aligned}&\sum_{x\in A} \mathbb{E}\left[\exp\left(t \Big| \sum_{i=1}^m \sigma_i x_i \Big| \right) \right] \\ &\quad\le \sum_{x\in A} \mathbb{E}\left[\exp\left(t \sum_{i=1}^m \sigma_i x_i \right) \right] + \sum_{x\in A} \mathbb{E}\left[\exp\left(- t \sum_{i=1}^m \sigma_i x_i \right) \right] \\ &\quad= 2 \sum_{x\in A} \prod_{i=1}^m \frac{\exp(t\,x_i) + \exp(-t\,x_i)}{2} \le 2 \sum_{x\in A} \prod_{i=1}^m \exp\left(\frac{t^2 x_i^2}{2} \right) \\ &\quad= 2 \sum_{x\in A} \exp\left(\frac{t^2}{2} \sum_{i=1}^m x_i^2 \right) \le 2 \sum_{x\in A} \exp\left(\frac{t^2 r^2}{2} \right) = 2|A| e^{\frac{t^2 r^2}{2}}.\end{aligned}$$

Taking the log of both sides and dividing by t, we have

$$\mathbb{E}\left[\sup_{x\in A} \sum_{i=1}^m \sigma_i x_i \right] \le \frac{\log(2|A|)}{t} + \frac{t r^2}{2}.$$

Choosing $t = \frac{\sqrt{2\log(2|A|)}}{r}$ concludes the proof.

Appendix C. Detailed Optimization Error Analysis

This section provides a detailed expansion of Section 3.3 in the main text. In this section, Building upon and further developing the technique proposed in Kohler and Krzyzak (2023a), we conduct an in-depth analysis of the optimization error bounds, addressing both iteration error and initialization error separately. Proofs for auxiliary lemmas are also included. Notation specific to this section is shown in Table 2.

Table 2: Notation specific to this section

$\bar{u} = u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$	the best approximation element defined in Equation 13
$(\bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_{\bar{\mathfrak{m}}})$	the parameters of the $\bar{\mathfrak{m}}$ sub-networks in $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$
$(\bar{c}_1, \ldots, \bar{c}_{\bar{\mathfrak{m}}})$	the linear coefficients of $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}$
$(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}$	the random sub-network initialization $(\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\mathfrak{m}})^{[0]}$
$\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}$	$(c_1^*, \ldots, c_{\mathfrak{m}}^*)$, where c_i^* is defined in Equations 17-19
$\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*}$	the transition parameters $\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*} = ((\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}})^{[0]}, \boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*})$
$u_{\mathfrak{m}}^*$	the $\mathcal{PNN}$ parameterized by $\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m},*}$
R	a parameter controlling the iteration error
Q	a parameter controlling the initialization error with high probability

C.1 Analysis of the Iteration Error

We first introduce the following results for the PGD algorithm, with the complete proof provided in Appendix C.3.

Lemma 20 *Let $d_1, d_2 \in \mathbb{N}$, let $U, V, K \geq 0$, let $\mathrm{X} \subset \mathbb{R}^{d_1}$ and $\mathrm{Y} \subseteq \mathbb{R}^{d_2}$ be closed and convex, and let $F : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^+$ be a function such that $F(\boldsymbol{x}, \boldsymbol{y})$ is differentiable while $\boldsymbol{y} \mapsto F(\boldsymbol{x}, \boldsymbol{y})$ is convex for all $\boldsymbol{x} \in \mathbb{R}^{d_1}$. Assume that*

$$\|\nabla_{\boldsymbol{y}} F(\boldsymbol{x}, \boldsymbol{y})\|_2 \leq V\,, \tag{38}$$

$$\|\nabla F(\boldsymbol{x}_1, \boldsymbol{y}_1) - \nabla F(\boldsymbol{x}_2, \boldsymbol{y}_2)\|_2 \leq K\|(\boldsymbol{x}_1, \boldsymbol{y}_1) - (\boldsymbol{x}_2, \boldsymbol{y}_2)\|_2 \tag{39}$$

for all $(\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2) \in \mathrm{X} \times \mathrm{Y}$. Choose $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathrm{X} \times \mathrm{Y}$ and set

$$(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) = \mathrm{Proj}_{\mathrm{X}\times\mathrm{Y}}\big\{(\boldsymbol{x}_t, \boldsymbol{y}_t) - \lambda \nabla F(\boldsymbol{x}_t, \boldsymbol{y}_t)\big\} \tag{40}$$

for $t = 0, 1, \ldots, T$, where

$$\lambda = \min\{T^{-1}, 2K^{-1}\}\,.$$

Let $\boldsymbol{y}^* \in Y$ and assume

$$|F(\boldsymbol{x}_t, \boldsymbol{y}^*) - F(\boldsymbol{x}_0, \boldsymbol{y}^*)| \leq U\|\boldsymbol{y}^*\|_2 \|\boldsymbol{x}_t - \boldsymbol{x}_0\|_2 \tag{41}$$

for all $t = 1, \dots, T$. Then it holds:

$$F(\boldsymbol{x}_T, \boldsymbol{y}_T) - F(\boldsymbol{x}_0, \boldsymbol{y}^*) \leq U\|\boldsymbol{y}^*\|_2 \operatorname{diam}(\mathrm{X}) + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}_0\|_2^2}{2} + \frac{V^2}{2T}\,. \tag{42}$$

To apply Lemma 20 to our setting, we identify F with $\widehat{F}$, $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ with $(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}})^{[t]}$, and $\boldsymbol{y}^*$ with $\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m},*}$, which directly corresponds to the iteration error in Equation 22. This approach requires satisfying conditions 38, 39, and 41. The following lemmas characterize the relevant properties of $\widehat{F}$, with proofs provided in Appendices C.4-C.7. We begin by establishing an upper bound for $\nabla_{\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}} \widehat{F}(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}})$ to satisfy condition 38.

Lemma 21 *For $u_{\mathfrak{m},\boldsymbol{\theta}} \in \mathcal{PNN}(\mathfrak{m}, M, \{W, L, B_{\boldsymbol{\theta}}\})$, denote the empirical risk $\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})$ from Equation 7 as $\widehat{F}(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}}) = \widehat{F}(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}})$, omitting the dependence on sample points. Then*

$$\left\|\nabla_{\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}} \widehat{F}\big(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}\big)\right\|_2^2 \leq C_1 \mathfrak{m} M^2 B_{\boldsymbol{\theta}}^{4L}\,, \tag{43}$$

where C_1 is a universal constant depending only on Ω, W, L, d and B_0.

To satisfy Equation 39, we estimate the Lipschitz constant of $\nabla \widehat{F}(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}})$. We first associate the Lipschitz property of the gradient ∇f with the norm of Hessian matrix $\nabla^2 f$.

Lemma 22 *For $f(\boldsymbol{x})$ convex and twice differentiable, it holds that*

$$\|\nabla^2 f(\boldsymbol{x})\|_{2,2} \leq \|\nabla^2 f(\boldsymbol{x})\|_{\mathrm{F}} \leq K \Longrightarrow \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq K\|\boldsymbol{x} - \boldsymbol{y}\|_2\,,$$

where $\|\cdot\|_{2,2}$ is the spectral norm of the matrix, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of the matrix.

Thus, it suffices to estimate the Frobenius norm of the Hessian matrix of $\widehat{F}(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}})$.

Lemma 23 *With notation consistent with Lemma 21, we have*

$$\left\|\nabla^2_{\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}}} \widehat{F}\big(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}}\big)\right\|_{\mathrm{F}} \leq C_2 \mathfrak{m} M^2 B_{\boldsymbol{\theta}}^{4L}\,,$$

where C_2 is a universal constant which only depends on Ω, W, L, d and B_0. Using Lemma 22, $\nabla \widehat{F}(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}})$ is then equipped with Lipschitz constant $C_2 \mathfrak{m} M^2 B_{\boldsymbol{\theta}}^{4L}$.

Finally, we assures that $\widehat{F}$ satisfies Equation 41.

Lemma 24 *With notation consistent with Lemma 21, we have*

$$\left|\widehat{F}\big(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},1}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}\big) - \widehat{F}\big(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},2}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}\big)\right| \leq C_3 M B_{\boldsymbol{\theta}}^{3L} \big\|\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}\big\|_2 \big\|\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},1} - \boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},2}\big\|_2\,,$$

where $\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},1}$ and $\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m},2}$ denote two different sub-network weight vectors. C_3 is a universal constant which only depends on Ω, W, L, d and B_0.

Let $\zeta = \bar{M}$ in Equation 11 to ensure $\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m},*} \in B_\zeta = B_{\bar{M}}$, while we naturally have $(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}})^{[0]} \in A_\eta$. Combining Lemmas 20-24, we achieve the following corollary to bound the iteration error.

Corollary 1 (Proposition 2 in the main text) *Let $\zeta = \bar{M}$ in Equation 11. Then, the output $u_{\mathcal{A}}$ of the PGD algorithm in Equation 12 belongs to $\mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$. When we run the algorithm with step size*

$$\lambda = \min\{T^{-1}, 2C_2^{-1}\mathfrak{m}^{-1}\bar{M}^{-2}(B_{\bar{\boldsymbol{\theta}}} + \eta)^{-4\bar{L}}\},$$

where T is the total number of iterations and η is the projection radius in Equation 10, the iteration error in Equation 22 satisfies

$$\begin{aligned}\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(u_{\mathfrak{m}}^*) &\leq C_3\bar{M}\eta(B_{\bar{\boldsymbol{\theta}}} + \eta)^{3\bar{L}}\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_2 + \frac{1}{2}\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_2^2 + \frac{C_1\mathfrak{m}\bar{M}^2(B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{2T} \\ &\leq \frac{C_3\bar{M}^2(B_{\bar{\boldsymbol{\theta}}} + \eta)^{3\bar{L}}\eta}{\sqrt{R}} + \frac{\bar{M}^2}{2R} + \frac{C_1\mathfrak{m}\bar{M}^2(B_{\bar{\boldsymbol{\theta}}} + \eta)^{4\bar{L}}}{2T}.\end{aligned}$$

Here, $u_{\mathfrak{m}}^* \in \mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ is defined in Equation 20, C_1 is from Lemma 21, C_2 is from Lemma 23, and C_3 is from Lemma 24.

Remark 11 *In Corollary 1, we use the following property of $\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_2$:*

$$\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_2 = \sqrt{\sum_{s=1}^{\mathfrak{m}} |c_s^*|^2} = \frac{1}{\sqrt{R}}\sqrt{\sum_{k=1}^{\bar{\mathfrak{m}}} |\bar{c}_k|^2} \leq \frac{1}{\sqrt{R}}\sum_{k=1}^{\bar{\mathfrak{m}}} |\bar{c}_k| = \frac{1}{\sqrt{R}}\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_1 \leq \frac{\bar{M}}{\sqrt{R}}. \tag{44}$$

That is, as R, which controls the over-parameterization degree of $u_{\mathfrak{m},\boldsymbol{\theta}}$, increases, the upper bound of $\|\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m},*}\|_2$ decays polynomially. As we have seen, this property allows us to control the iteration error to any given precision by letting $R \to \infty$ with $\bar{M}$ fixed, which underscores the importance of over-parameterization in our analysis.

C.2 Analysis of the Initialization Error

We first establish the following lemma to estimate the probability of $G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$, with proof provided in Appendix C.8.

Lemma 25 *Consider $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ from Theorem 3 with sub-network parameters $(\bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_{\bar{\mathfrak{m}}})$. If*

$$\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q, \quad R, Q \in \mathbb{N} \text{ and } Q \text{ is sufficiently large},$$

then, it holds that

$$\mathbb{P}\big(G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\big) \geq 1 - \bar{\mathfrak{m}}R\Big[1 - \delta^{\bar{W}(\bar{W}+1)\bar{L}}(2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^Q.$$

Intuitively, the initialization error can be effectively controlled when the target weights $\bar{\boldsymbol{\theta}}_k$ and random initialization $(\boldsymbol{\theta}_{s_{k,v}})^{[0]}$ exhibit only minimal differences. The following lemma formalizes this relationship, with proof provided in Appendix C.9.

Lemma 26 *For $u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}} \in \mathcal{PNN}(\bar{\mathfrak{m}}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}}\})$ in Theorem 3, we have*

$$\widehat{\mathcal{L}}(u^*_{\mathfrak{m}}) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}) \leq C_4 \bar{M}^2 B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}} \max_{k=1,\ldots,\bar{\mathfrak{m}}} \max_{v=1,\ldots,R} \left\| (\boldsymbol{\theta}_{s_{k,v}})^{[0]} - \bar{\boldsymbol{\theta}}_k \right\|_\infty$$

where $u^*_{\mathfrak{m}} \in \mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$ is defined in Equation 20, the random indices $s_{k,v}$ are from Equation 16, and C_4 is a universal constant depending only on $\Omega, \bar{W}, \bar{L}, d$ and B_0.

Combining Lemmas 25 and 26, we can bound the initialization error with arbitrarily high probability and precision through proper selection of the sub-network size $\mathfrak{m}$ in $u_{\mathfrak{m},\boldsymbol{\theta}}$.

Proposition 5 (Proposition 3 in the main text**)** *For any $\delta > 0$ and $R, Q \in \mathbb{N}$ with Q sufficiently large, if we set $\mathfrak{m} = \bar{\mathfrak{m}} \cdot R \cdot Q$, then with probability at least*

$$1 - \bar{\mathfrak{m}} R \Big[1 - \delta^{\bar{W}(\bar{W}+1)\bar{L}} (2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}} \Big]^Q ,$$

the initialization error in Equation 22 satisfies

$$\widehat{\mathcal{L}}(u^*_{\mathfrak{m}}) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}) \leq C_4 \bar{M}^2 B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}} \delta \,,$$

where $u^*_{\mathfrak{m}} \in \mathcal{PNN}(\mathfrak{m}, \bar{M}, \{\bar{W}, \bar{L}, B_{\bar{\boldsymbol{\theta}}} + \eta\})$ is defined in Equation 20. C_4 is from Lemma 26.

C.3 Proof of Lemma 20

Note that $\boldsymbol{y}^* \in \mathrm{Y}$. By convexity of $\boldsymbol{y} \mapsto F(\boldsymbol{x}_t, \boldsymbol{y})$ and Equation 38, we have

$$\begin{aligned}
F(\boldsymbol{x}_t, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}^*) &\leq \left\langle \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{y}_t - \boldsymbol{y}^* \right\rangle \\
&= \frac{1}{2\lambda} \cdot 2 \cdot \left\langle \lambda \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{y}_t - \boldsymbol{y}^* \right\rangle \\
&= \frac{1}{2\lambda} \Big[- \left\| \boldsymbol{y}_t - \boldsymbol{y}^* - \lambda \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t) \right\|_2^2 + \|\boldsymbol{y}_t - \boldsymbol{y}^*\|_2^2 + \left\| \lambda \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t) \right\|_2^2 \Big] \\
&\leq \frac{1}{2\lambda} \Big[\|\boldsymbol{y}_t - \boldsymbol{y}^*\|^2 - \|\boldsymbol{y}_{t+1} - \boldsymbol{y}^*\|_2^2 + \lambda^2 V^2 \Big] .
\end{aligned}$$

Since $\lambda \leq T^{-1}$, this implies

$$\frac{1}{T} \sum_{t=0}^{T-1} \big[F(\boldsymbol{x}_t, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}^*) \big] \leq \frac{\|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2}{2} + \frac{V^2}{2T} .$$

Using above results and by Equation 41, we have

$$\begin{aligned}
\min_{t=0,\ldots,T} F(\boldsymbol{x}_t, \boldsymbol{y}_t) &\leq \frac{1}{T} \sum_{t=0}^{T-1} F(\boldsymbol{x}_t, \boldsymbol{y}^*) + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2}{2} + \frac{V^2}{2T} \\
&\leq F(\boldsymbol{x}_0, \boldsymbol{y}^*) + \frac{1}{T} \sum_{t=0}^{T-1} |F(\boldsymbol{x}_t, \boldsymbol{y}^*) - F(\boldsymbol{x}_0, \boldsymbol{y}^*)| + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}_0\|_2^2}{2} + \frac{V^2}{2T} . \\
&\leq F(\boldsymbol{x}_0, \boldsymbol{y}^*) + U \|\boldsymbol{y}^*\|_2 \operatorname{diam}(\mathrm{X}) + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}_0\|_2^2}{2} + \frac{V^2}{2T}
\end{aligned}$$

Denote by $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$. Since $\mathrm{X} \times \mathrm{Y}$ is a convex set, we have

$$\begin{aligned} F(\boldsymbol{z}_{t+1}) &= F(\boldsymbol{z}_t) + \int_0^1 \frac{\partial F\big(\boldsymbol{z}_t + w(\boldsymbol{z}_{t+1} - \boldsymbol{z}_t)\big)}{\partial w} \,\mathrm{d}w \\ &= F(\boldsymbol{z}_t) + \int_0^1 \nabla F\big(\boldsymbol{z}_t + w(\boldsymbol{z}_{t+1} - \boldsymbol{z}_t)\big)^{\mathrm{T}} (\boldsymbol{z}_{t+1} - \boldsymbol{z}_t) \,\mathrm{d}w \,. \end{aligned}$$

Since $\nabla F(\boldsymbol{x}, \boldsymbol{y})$ satisfies Equation 39, it holds that

$$\begin{aligned} &\int_0^1 \big[\nabla F\big(\boldsymbol{z}_t + w(\boldsymbol{z}_{t+1} - \boldsymbol{z}_t)\big) - \nabla F(\boldsymbol{z}_t)\big]^{\mathrm{T}} (\boldsymbol{z}_{t+1} - \boldsymbol{z}_t) \,\mathrm{d}w \\ &\qquad \le \int_0^1 K \|w(\boldsymbol{z}_{t+1} - \boldsymbol{z}_t)\|_2 \, \|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|_2 \,\mathrm{d}w = \frac{K}{2} \|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|^2 \,. \end{aligned}$$

By Equation 40, $\boldsymbol{z}_{t+1} = \mathrm{Proj}_{\mathrm{X} \times \mathrm{Y}}\big\{\boldsymbol{z}_t - \lambda \cdot \nabla F(\boldsymbol{z}_t)\big\}$, which implies

$$\big\langle \boldsymbol{z}_t - \lambda \cdot \nabla F(\boldsymbol{z}_t) - \boldsymbol{z}_{t+1} \,,\ \boldsymbol{u} - \boldsymbol{z}_{t+1} \big\rangle \le 0 \,, \qquad \forall\, \boldsymbol{u} \in \mathrm{X} \times \mathrm{Y} \,.$$

Let $\boldsymbol{u} = \boldsymbol{z}_t$, then $\nabla F(\boldsymbol{z}_t)^{\mathrm{T}} (\boldsymbol{z}_{t+1} - \boldsymbol{z}_t) \le -\frac{1}{\lambda} \|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|^2$. Thus, we get

$$\begin{aligned} F(\boldsymbol{z}_{t+1}) &\le F(\boldsymbol{z}_t) + \nabla F(\boldsymbol{z}_t)^{\mathrm{T}} (\boldsymbol{z}_{t+1} - \boldsymbol{z}_t) + \frac{K}{2} \|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|^2 \\ &\le F(\boldsymbol{z}_t) - \left(\frac{1}{\lambda} - \frac{K}{2}\right) \|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|^2 \,. \end{aligned}$$

Therefore, when $\lambda = T^{-1} \wedge 2K^{-1}$, it holds that

$$F(\boldsymbol{x}_T, \boldsymbol{y}_T) = \min_{t=0,\ldots,T} F(\boldsymbol{x}_t, \boldsymbol{y}_t) \le F(\boldsymbol{x}_0, \boldsymbol{y}^*) + U\|\boldsymbol{y}^*\| \operatorname{diam}(\mathrm{X}) + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2}{2} + \frac{V^2}{2T} \,.$$

C.4 Proof of Lemma 21

By Equation 7, $u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k(\boldsymbol{x})$, $B_0 = \max\{\|h\|_{L^\infty(\Omega)}, \|g\|_{L^\infty(\partial\Omega)}, \|w\|_{L^\infty(\Omega)}\}$ and $\sum_k |c_k| \le M$, it holds that

$$\begin{aligned} &\big\|\nabla_{\boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}} \widehat{F}\big(\boldsymbol{\theta}_{\mathrm{in}}^{\mathfrak{m}}, \boldsymbol{\theta}_{\mathrm{out}}^{\mathfrak{m}}\big)\big\|_2^2 = \bigg\|\nabla_{(c_k)_{k=1}^{\mathfrak{m}}} \widehat{\mathcal{L}}\bigg(\sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k\bigg)\bigg\|_2^2 \\ &\quad = \sum_{j=1}^{\mathfrak{m}} \bigg\{ \frac{|\Omega|}{N_{\mathrm{in}}} \sum_{p=1}^{N_{\mathrm{in}}} \bigg[\sum_{k=1}^{\mathfrak{m}} c_k \nabla_{\boldsymbol{x}} \phi_{\boldsymbol{\theta}}^k(X_p)^{\mathrm{T}} \nabla_{\boldsymbol{x}} \phi_{\boldsymbol{\theta}}^j(X_p) + w(X_p) \sum_{k=1}^{\mathfrak{m}} c_k \, \phi_{\boldsymbol{\theta}}^k(X_p) \, \phi_{\boldsymbol{\theta}}^j(X_p) \\ &\qquad\qquad - \phi_{\boldsymbol{\theta}}^j(X_p) h(X_p) \bigg] - \frac{|\partial\Omega|}{N_b} \sum_{q=1}^{N_b} \big[\phi_{\boldsymbol{\theta}}^j(Y_q) g(Y_q)\big] \bigg\}^2 \\ &\quad \le \sum_{j=1}^{\mathfrak{m}} 4 \bigg\{ \bigg\{ \frac{|\Omega|}{N_{\mathrm{in}}} \sum_{p=1}^{N_{\mathrm{in}}} \bigg[\sum_{k=1}^{\mathfrak{m}} c_k \nabla \phi_{\boldsymbol{\theta}}^k(X_p)^{\mathrm{T}} \nabla \phi_{\boldsymbol{\theta}}^j(X_p) \bigg] \bigg\}^2 \\ &\qquad\qquad + \bigg\{ \frac{|\Omega|}{N_{\mathrm{in}}} \sum_{p=1}^{N_{\mathrm{in}}} \bigg[w(X_p) \sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k(X_p) \phi_{\boldsymbol{\theta}}^j(X_p) \bigg] \bigg\}^2 \end{aligned}$$

$$
\begin{aligned}
&\quad + \left\{ \frac{|\Omega|}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \left[\phi_{\boldsymbol{\theta}}^j(X_p) h(X_p) \right] \right\}^2 + \left\{ \frac{|\partial\Omega|}{N_b} \sum_{q=1}^{N_b} [\phi_{\boldsymbol{\theta}}^j(Y_q) g(Y_q)] \right\}^2 \Bigg\} \\
&\le \sum_{j=1}^{\mathfrak{m}} 4 \Bigg\{ \left\{ |\Omega|^2 \frac{1}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \Big\| \sum_{k=1}^{\mathfrak{m}} c_k \nabla \phi_{\boldsymbol{\theta}}^k(X_p) \Big\|_2^2 \cdot \frac{1}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \big\| \nabla \phi_{\boldsymbol{\theta}}^j(X_p) \big\|_2^2 \right\} \\
&\qquad + \left\{ |\Omega|^2 \frac{1}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \Big| w(X_p) \sum_{k=1}^{\mathfrak{m}} c_k \phi_{\boldsymbol{\theta}}^k(X_p) \Big|^2 \cdot \frac{1}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \big| \phi_{\boldsymbol{\theta}}^j(X_p) \big|^2 \right\} \\
&\qquad + (|\Omega|^2 + |\partial\Omega|^2) B_0^2 \max_{\boldsymbol{x}} |\phi_{\boldsymbol{\theta}}^j(\boldsymbol{x})|^2 \Bigg\} \\
&\le 4\mathfrak{m} \Big\{ |\Omega|^2 M^2 \max_{k,\boldsymbol{x}} \big\| \nabla \phi_{\boldsymbol{\theta}}^k(\boldsymbol{x}) \big\|_2^2 \max_{j,\boldsymbol{x}} \big\| \nabla \phi_{\boldsymbol{\theta}}^j(\boldsymbol{x}) \big\|_2^2 + (|\Omega|^2 + |\partial\Omega|^2) B_0^2 \max_{j,\boldsymbol{x}} |\phi_{\boldsymbol{\theta}}^j(\boldsymbol{x})|^2 \\
&\qquad + |\Omega|^2 B_0^2 M^2 \max_{k,\boldsymbol{x}} \big| \phi_{\boldsymbol{\theta}}^k(\boldsymbol{x}) \big|^2 \max_{j,\boldsymbol{x}} \big| \phi_{\boldsymbol{\theta}}^j(\boldsymbol{x}) \big|^2 \Big\} ,
\end{aligned}
$$

where the Cauchy-Schwarz inequality is used in the third step. By Lemma 9, we have

$$
|\phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \le (W+1) B_{\boldsymbol{\theta}} , \qquad |\partial_{x_m} \phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \le W^{L-1} B_{\boldsymbol{\theta}}^L .
$$

Then, we get $\max_{k,\boldsymbol{x}} \|\nabla \phi_{\boldsymbol{\theta}}^k(\boldsymbol{x})\|_2^2 \le d\big(W^{L-1} B_{\boldsymbol{\theta}}^L\big)^2$, which concludes the proof.

C.5 Proof of Lemma 22

Write $\boldsymbol{x} = \sum_{j=1}^n a_j \boldsymbol{e}_j$. Suppose $\|\boldsymbol{x}\|_2 = 1$, that is, $\sum_j |a_j|^2 = 1$. Then, it holds that

$$
\|\boldsymbol{A}\boldsymbol{x}\|_2^2 = \Big\| \sum_{j=1}^n a_j \boldsymbol{A}\boldsymbol{e}_j \Big\|_2^2 \le \Big(\sum_j |a_j| \|\boldsymbol{A}\boldsymbol{e}_j\|_2 \Big)^2 \le \Big(\sum_{j=1}^n |a_j|^2 \Big) \sum_{j=1}^n \|A\boldsymbol{e}_j\|_2^2 = \|\boldsymbol{A}\|_{\rm F}^2 .
$$

Since $\boldsymbol{x}$ is arbitrary, we get $\|\boldsymbol{A}\|_2 \le \|\boldsymbol{A}\|_{\rm F}$. Thus, if we have $\|\nabla^2 f(\boldsymbol{x})\|_{2,2} \le \|\nabla^2 f(\boldsymbol{x})\|_{\rm F} \le K$, it further holds that $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \le \|\nabla^2 f(\boldsymbol{\xi})\|_{2,2} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le K \|\boldsymbol{x} - \boldsymbol{y}\|_2$.

C.6 Proof of Lemma 23

The goal is to estimate the Frobenius norm of the Hessian matrix of $\widehat{F}(\boldsymbol{\theta}_{\rm total}^{\mathfrak{m}})$, which is equivalent to estimate the second derivative of $\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})$ w.r.t. the weight parameter. For simplicity, we will demonstrate the proof process using the derivative w.r.t. $a_{1,1,1}^{(0)}$, the innermost weight of the first sub-network in $u_{\mathfrak{m},\boldsymbol{\theta}}$. Denote by $L^\infty := L^\infty(\bar{\Omega})$. We have

$$
\begin{aligned}
&\Big| \partial_{a_{1,1,1}^{(0)}} \partial_{a_{1,1,1}^{(0)}} \widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}}) \Big| \\
&\quad \le \left| \partial_{a_{1,1,1}^{(0)}} \partial_{a_{1,1,1}^{(0)}} \frac{|\Omega|}{N_{\rm in}} \sum_{p=1}^{N_{\rm in}} \left[\frac{1}{2} \sum_{i=1}^d |\partial_{x_i} u_{\mathfrak{m},\boldsymbol{\theta}}(X_p)|^2 + \frac{1}{2} w(X_p) u_{\mathfrak{m},\boldsymbol{\theta}}^2(X_p) - u_{\mathfrak{m},\boldsymbol{\theta}}(X_p) h(X_p) \right] \right| \\
&\qquad + \left| \partial_{a_{1,1,1}^{(0)}} \partial_{a_{1,1,1}^{(0)}} \frac{|\partial\Omega|}{N_b} \sum_{q=1}^{N_b} \left[u_{\mathfrak{m},\boldsymbol{\theta}}(Y_q) g(Y_q) \right] \right|
\end{aligned}
$$

$$\lesssim \Big\| \sum_{i=1}^{d} \big[\partial_{a^{(0)}_{1,1,1}} \partial_{x_i} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \cdot \partial_{a^{(0)}_{1,1,1}} \partial_{x_i} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) + \partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} \partial_{x_i} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \cdot \partial_{x_i} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \big] \Big\|_{L^\infty}$$
$$+ \big\| \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}} \cdot \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}} + u_{\mathfrak{m},\boldsymbol{\theta}} \cdot \partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}} \big\|_{L^\infty} + \big\| \partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}} \big\|_{L^\infty} ,$$

Therefore, the task reduces to estimating the following partial derivatives:

- **First Order Derivatives:** $\partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$, $\partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$.
- **Second Order Derivatives:** $\partial_{a^{(0)}_{1,1,1}} \partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$, $\partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$.
- **Third Order Derivative:** $\partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} \partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$.

In the rest of the proof, we will use $\phi^{(\ell)}_{k,j}(\boldsymbol{x})$ to denote the j-th output of the k-th sub-network at layer ℓ. Also, we assume all weights are non-negative when estimating the upper bounds of the partial derivatives (if not, the usual triangle inequality yields the same conclusion).

We first focus on the first order derivatives. It holds that

$$\partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{\mathfrak{m}} c_k \cdot \partial_{x_1} \phi^k_{\boldsymbol{\theta}}(\boldsymbol{x}) .$$

Also, we have

$$\begin{aligned}
\partial_{x_1} \phi^k_{\boldsymbol{\theta}}(\boldsymbol{x}) = \partial_{x_1} \phi^{(L)}_{k,1}(\boldsymbol{x}) &= \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{k,1,s_{L-1}} \partial_{x_1} \phi^{(L-1)}_{k,s_{L-1}}(\boldsymbol{x}) \\
&= \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{k,1,s_{L-1}} \rho' \bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{k,1,s_{L-2}} \cdot \phi^{(L-2)}_{k,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{k,s_{L-1}} \bigg) \cdot \\
&\qquad \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{k,s_{L-1},s_{L-2}} \cdot \partial_{x_1} \phi^{(L-2)}_{k,s_{L-2}}(\boldsymbol{x}) \bigg] \\
&\le \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{k,1,s_{L-1}} \cdot 1 \cdot \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{k,s_{L-1},s_{L-2}} \cdot 1 \cdot \bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{k,s_{L-2},s_{L-3}} \cdots \sum_{s_1=1}^{N_1} a^{(1)}_{k,s_2,s_1} \cdot a^{(0)}_{k,s_2,1} \bigg] \bigg] \\
&\le \sum_{s_{L-1}=1}^{N_{L-1}} \sum_{s_{L-2}=1}^{N_{L-2}} \cdots \sum_{s_1=1}^{N_1} a^{(L-1)}_{k,1,s_{L-1}} a^{(L-2)}_{k,s_{L-1},s_{L-2}} \cdots a^{(1)}_{k,s_2,s_1} \cdot a^{(0)}_{k,s_2,1} \le \bigg(\prod_{i=1}^{L-1} N_i \bigg) (B_{\boldsymbol{\theta}})^L .
\end{aligned}$$

Therefore, it holds that

$$\partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \le M \bigg(\prod_{i=1}^{L-1} N_i \bigg) (B_{\boldsymbol{\theta}})^L .$$

As for the partial derivative w.r.t. $a^{(0)}_{1,1,1}$, we have

$$\partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = c_1 \cdot \partial_{a^{(0)}_{1,1,1}} \phi^1_{\boldsymbol{\theta}}(\boldsymbol{x}) . \tag{45}$$

Also, it holds that

$$\begin{aligned}
\partial_{a^{(0)}_{1,1,1}}\phi^1_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \partial_{a^{(0)}_{1,1,1}}\phi^{(L)}_{k,1}(\boldsymbol{x}) = \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}}\partial_{a^{(0)}_{1,1,1}}\phi^{(L-1)}_{1,s_{L-1}}(\boldsymbol{x}) \\
&= \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}}\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}}\bigg)\cdot \\
&\qquad \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{a^{(0)}_{1,1,1}}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\bigg] \\
&= \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}}\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}}\bigg)\cdot \\
&\qquad \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\bigg[\cdots\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1}\cdot x_1\rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0} + b^{(0)}_{1,s_1}\bigg)\bigg]\bigg] \\
&= \sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}}\cdot\bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdots\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1}\cdot x_1\bigg]\bigg]\cdot \\
&\quad \rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}}\bigg)\cdots\rho'\bigg(\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1}\cdot\phi^{(1)}_{1,s_1}(\boldsymbol{x}) + b^{(1)}_{1,s_2}\bigg)\cdot \\
&\quad \rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0} + b^{(0)}_{1,s_1}\bigg) \le \bigg(\prod_{i=1}^{L-1} N_i\bigg)(B_{\boldsymbol{\theta}})^{L-1}\,. \qquad (46)
\end{aligned}$$

Therefore, we have

$$\partial_{a^{(0)}_{1,1,1}}u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \le M\bigg(\prod_{i=1}^{L-1} N_i\bigg)(B_{\boldsymbol{\theta}})^{L-1}\,.$$

Next, we estimate the second order partial derivatives. By Equation 45, we have

$$\partial_{a^{(0)}_{1,1,1}}\partial_{x_1}u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = c_1\cdot\partial_{x_1}\partial_{a^{(0)}_{1,1,1}}\phi^1_{\boldsymbol{\theta}}(\boldsymbol{x})\,. \qquad (47)$$

Meanwhile, by Equation 46, it holds that

$$\begin{aligned}
&\partial_{x_1}\partial_{a^{(0)}_{1,1,1}}\phi^1_{\boldsymbol{\theta}}(\boldsymbol{x}) \\
&= \bigg[\sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}}\cdot\bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdots\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1}\bigg]\bigg]\bigg]\cdot \\
&\quad \rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}}\bigg)\cdots \\
&\quad \rho'\bigg(\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1}\cdot\phi^{(1)}_{1,s_1}(\boldsymbol{x}) + b^{(1)}_{1,s_2}\bigg)\cdot\rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0} + b^{(0)}_{1,s_1}\bigg)
\end{aligned}$$

$$
\begin{aligned}
&+\Bigg[\sum_{s_{L-1}=1}^{N_{L-1}} a_{1,1,s_{L-1}}^{(L-1)}\cdot\Bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\Bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a_{1,s_{L-2},s_{L-3}}^{(L-3)}\cdots\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot x_1\Bigg]\Bigg]\Bigg]\cdot\\
&\partial_{x_1}\Bigg[\rho'\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\cdot\\
&\cdots\rho'\Bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\Bigg)\cdot\rho'\Bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\Bigg)\Bigg]\\
\le&\Bigg(\prod_{i=1}^{L-1}N_i\Bigg)(B_{\boldsymbol{\theta}})^L\Bigg[\rho'\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\cdots\\
&\rho'\Bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\Bigg)\cdot\rho'\Bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\Bigg)\\
&+x_1\partial_{x_1}\rho'\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\\
&\cdots\rho'\Bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\Bigg)\cdot\rho'\Bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\Bigg)\cdot\\
&+\cdots+x_1\rho'\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\cdot\\
&\cdots\rho'\Bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\Bigg)\cdot\partial_{x_1}\rho'\Bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\Bigg)\Bigg]. \qquad (48)
\end{aligned}
$$

Since we have

$$
\begin{aligned}
&\partial_{x_1}\rho'\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\\
&\quad=\rho''\Bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,1,s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\Bigg)\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\partial_{x_1}\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})\\
&\quad\le\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\Bigg(\prod_{i=1}^{L-3}N_i\Bigg)(B_{\boldsymbol{\theta}})^{L-2}\le\Bigg(\prod_{i=1}^{L-2}N_i\Bigg)(B_{\boldsymbol{\theta}})^{L-1}.
\end{aligned}
$$

Then, above estimations lead to

$$
\partial_{a_{1,1,1}^{(0)}}\partial_{x_1}\phi_{\boldsymbol{\theta}}^1(\boldsymbol{x})\le\Bigg(\prod_{i=1}^{L-1}N_i\Bigg)(B_{\boldsymbol{\theta}})^L\cdot L\cdot\Bigg(\prod_{i=1}^{L-1}N_i\Bigg)(B_{\boldsymbol{\theta}})^L=L\cdot\Bigg(\prod_{i=1}^{L-1}N_i\Bigg)^2\cdot(B_{\boldsymbol{\theta}})^{2L}.
$$

Therefore, it holds that

$$
\partial_{a_{1,1,1}^{(0)}}\partial_{x_1}u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\le M\cdot L\cdot\Bigg(\prod_{i=1}^{L-1}N_i\Bigg)^2\cdot(B_{\boldsymbol{\theta}})^{2L}.
$$

Besides, w.r.t $a^{(0)}_{1,1,1}$, we have

$$\begin{aligned}
&\partial_{a^{(0)}_{1,1,1}} \rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}} \bigg) \\
&\quad = \rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-1)}_{1,s_{L-1}} \bigg) \sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \partial_{a^{(0)}_{1,1,1}} (\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})) \\
&\leq \sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \bigg(\prod_{i=1}^{L-3} N_i \bigg) (B_{\boldsymbol{\theta}})^{L-3} \leq \bigg(\prod_{i=1}^{L-2} N_i \bigg) (B_{\boldsymbol{\theta}})^{L-2} .
\end{aligned}$$

Then, it holds that

$$\partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) \leq M \cdot L \cdot \bigg(\prod_{i=1}^{L-1} N_i \bigg)^2 \cdot (B_{\boldsymbol{\theta}})^{2L-1} .$$

Finally, we turn to the third order derivative. Initially, by Equation 47, we have

$$\partial_{a^{(0)}_{1,1,1}} \partial_{a^{(0)}_{1,1,1}} \partial_{x_1} u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x}) = c_1 \cdot \partial_{a^{(0)}_{1,1,1}} \partial_{x_1} \partial_{a^{(0)}_{1,1,1}} \phi^1_{\boldsymbol{\theta}}(\boldsymbol{x}) .$$

Then, by fully expanding Equation 48, it holds that

$$\begin{aligned}
&\partial_{x_1} \partial_{a^{(0)}_{1,1,1}} \phi^1_{\boldsymbol{\theta}}(\boldsymbol{x}) \\
&\quad = \bigg[\sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}} \cdot \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}} \cdots \sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1} \bigg] \bigg] \bigg] \cdot \\
&\qquad \bigg[\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}} \bigg) \cdots \\
&\qquad \rho'\bigg(\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1} \cdot \phi^{(1)}_{1,s_1}(\boldsymbol{x}) + b^{(1)}_{1,s_2} \bigg) \cdot \rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0} \cdot x_{s_0} + b^{(0)}_{1,s_1} \bigg) \\
&\qquad + x_1 \partial_{x_1} \rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}} \bigg) \cdot \\
&\qquad \cdots \rho'\bigg(\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1} \cdot \phi^{(1)}_{1,s_1}(\boldsymbol{x}) + b^{(1)}_{1,s_2} \bigg) \cdot \rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0} \cdot x_{s_0} + b^{(0)}_{1,s_1} \bigg) \\
&\qquad + \cdots + x_1 \rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x}) + b^{(L-2)}_{1,s_{L-1}} \bigg) \cdot \\
&\qquad \cdots \rho'\bigg(\sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1} \cdot \phi^{(1)}_{1,s_1}(\boldsymbol{x}) + b^{(1)}_{1,s_2} \bigg) \cdot \partial_{x_1} \rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0} \cdot x_{s_0} + b^{(0)}_{1,s_1} \bigg) \bigg] \\
&\quad = \bigg[\sum_{s_{L-1}=1}^{N_{L-1}} a^{(L-1)}_{1,1,s_{L-1}} \cdot \bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}} \cdot \bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}} \cdots \sum_{s_1=1}^{N_1} a^{(1)}_{1,s_2,s_1} \bigg] \bigg] \bigg] \cdot
\end{aligned}$$

$$
\begin{aligned}
&\Bigg[\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\cdots\\
&\rho'\bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\bigg)\cdot\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\\
&+x_1\rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\partial_{x_1}\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})\\
&\cdot\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a_{1,s_{L-2},s_{L-3}}^{(L-3)}\cdot\phi_{1,s_{L-3}}^{(L-3)}(\boldsymbol{x})+b_{1,s_{L-2}}^{(L-3)}\bigg)\cdots\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\\
&+\cdots+x_1\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\cdot\\
&\cdots\rho'\bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\bigg)\cdot\partial_{x_1}\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\Bigg].
\end{aligned}
$$

Further taking derivative w.r.t. $a_{1,1,1}^{(0)}$, we have

$$
\begin{aligned}
&\partial_{a_{1,1,1}^{(0)}}\partial_{x_1}\partial_{a_{1,1,1}^{(0)}}\phi_{\boldsymbol{\theta}}^{1}(\boldsymbol{x})\\
&=\Bigg[\sum_{s_{L-1}=1}^{N_{L-1}} a_{1,1,s_{L-1}}^{(L-1)}\cdot\Bigg[\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\Bigg[\sum_{s_{L-3}=1}^{N_{L-3}} a_{1,s_{L-2},s_{L-3}}^{(L-3)}\cdots\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\Bigg]\Bigg]\Bigg]\cdot\\
&\quad\Bigg[\partial_{a_{1,1,1}^{(0)}}\Bigg[\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\cdots\\
&\quad\rho'\bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\bigg)\cdot\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\Bigg]\\
&\quad+x_1\partial_{a_{1,1,1}^{(0)}}\Bigg[\rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\cdot\\
&\quad\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\partial_{x_1}\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})\cdot\\
&\quad\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a_{1,s_{L-2},s_{L-3}}^{(L-3)}\cdot\phi_{1,s_{L-3}}^{(L-3)}(\boldsymbol{x})+b_{1,s_{L-2}}^{(L-3)}\bigg)\cdots\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\Bigg]\\
&\quad+\cdots+x_1\partial_{a_{1,1,1}^{(0)}}\Bigg[\rho'\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a_{1,s_{L-1},s_{L-2}}^{(L-2)}\cdot\phi_{1,s_{L-2}}^{(L-2)}(\boldsymbol{x})+b_{1,s_{L-1}}^{(L-2)}\bigg)\\
&\quad\cdots\rho'\bigg(\sum_{s_1=1}^{N_1} a_{1,s_2,s_1}^{(1)}\cdot\phi_{1,s_1}^{(1)}(\boldsymbol{x})+b_{1,s_2}^{(1)}\bigg)\cdot\partial_{x_1}\rho'\bigg(\sum_{s_0=1}^{d} a_{1,s_1,s_0}^{(0)}\cdot x_{s_0}+b_{1,s_1}^{(0)}\bigg)\Bigg]\Bigg].
\end{aligned}
$$

The calculation of the typical item tells us

$$
\begin{aligned}
&\partial_{a^{(0)}_{1,1,1}}\Bigg[\rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})+b^{(L-2)}_{1,s_{L-1}}\bigg)\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{x_1}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\\
&\qquad\cdot\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdot\phi^{(L-3)}_{1,s_{L-3}}(\boldsymbol{x})+b^{(L-3)}_{1,s_{L-2}}\bigg)\cdots\rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0}+b^{(0)}_{1,s_1}\bigg)\Bigg]\\
&=\rho'''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})+b^{(L-2)}_{1,s_{L-1}}\bigg)\\
&\quad\cdot\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{a^{(0)}_{1,1,1}}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{x_1}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\\
&\quad\cdot\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdot\phi^{(L-3)}_{1,s_{L-3}}(\boldsymbol{x})+b^{(L-3)}_{1,s_{L-2}}\bigg)\cdots\rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0}+b^{(0)}_{1,s_1}\bigg)\\
&\quad+\rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})+b^{(L-2)}_{1,s_{L-1}}\bigg)\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{a^{(0)}_{1,1,1}}\partial_{x_1}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\\
&\quad\cdot\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdot\phi^{(L-3)}_{1,s_{L-3}}(\boldsymbol{x})+b^{(L-3)}_{1,s_{L-2}}\bigg)\cdots\rho'\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0}+b^{(0)}_{1,s_1}\bigg)\\
&\quad+\cdots+\rho''\bigg(\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})+b^{(L-2)}_{1,s_{L-1}}\bigg)\sum_{s_{L-2}=1}^{N_{L-2}} a^{(L-2)}_{1,s_{L-1},s_{L-2}}\cdot\partial_{x_1}\phi^{(L-2)}_{1,s_{L-2}}(\boldsymbol{x})\\
&\quad\cdot\rho'\bigg(\sum_{s_{L-3}=1}^{N_{L-3}} a^{(L-3)}_{1,s_{L-2},s_{L-3}}\cdot\phi^{(L-3)}_{1,s_{L-3}}(\boldsymbol{x})+b^{(L-3)}_{1,s_{L-2}}\bigg)\cdots\rho''\bigg(\sum_{s_0=1}^{d} a^{(0)}_{1,s_1,s_0}\cdot x_{s_0}+b^{(0)}_{1,s_1}\bigg)\cdot x_1\\
&\le(2L+2)\bigg(\prod_{i=1}^{L-1}N_i\bigg)^2(B_{\boldsymbol{\theta}})^{2L}\,.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
&\partial_{a^{(0)}_{1,1,1}}\partial_{a^{(0)}_{1,1,1}}\partial_{x_1}u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\\
&\quad\le c_1\cdot\sum_{s_{L-1}=1}^{N_{L-1}}\sum_{s_{L-2}=1}^{N_{L-2}}\cdots\sum_{s_1=1}^{N_1} a^{(L-1)}_{k,1,s_{L-1}}a^{(L-2)}_{k,s_{L-1},s_{L-2}}\cdots a^{(1)}_{k,s_2,s_1}\cdot L\cdot(2L+2)\bigg(\prod_{i=1}^{L-1}N_i\bigg)^2(B_{\boldsymbol{\theta}})^{2L}\\
&\quad\le M\cdot L\cdot(2L+2)\bigg(\prod_{i=1}^{L-1}N_i\bigg)^3(B_{\boldsymbol{\theta}})^{3L-1}\,.
\end{aligned}
$$

Combining above estimations together, we obtain the total upper bound

$$
\Big|\partial_{a^{(0)}_{1,1,1}}\partial_{a^{(0)}_{1,1,1}}\widehat{\mathcal{L}}(u_{\mathfrak{m},\boldsymbol{\theta}})\Big|\le C(d,B_0,L)\cdot M^2\cdot\bigg(\prod_{i=1}^{L-1}N_i\bigg)^4(B_{\boldsymbol{\theta}})^{4L}\,.
$$

It can be readily observed that the upper bound obtained by differentiating the neural network output w.r.t. $a^{(0)}_{1,1,1}$ also controls the upper bounds obtained by differentiating the output w.r.t. general $a^{(\ell)}_{k,i,j}$, which concludes the proof.

C.7 Proof of Lemma 24

For $i = 1, 2$, when $u_{\mathfrak{m},\boldsymbol{\theta}}$ is parameterized with $(\boldsymbol{\theta}^{\mathfrak{m},i}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}})$, we denote it as $u_{\mathfrak{m},i} = \sum_{k=1}^{\mathfrak{m}} c_k \cdot \phi^k_{\boldsymbol{\theta},i}$. Then, it holds that

$$
\begin{aligned}
&\big|\widehat{F}\big(\boldsymbol{\theta}^{\mathfrak{m},1}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}\big) - \widehat{F}\big(\boldsymbol{\theta}^{\mathfrak{m},2}_{\text{in}}, \boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}\big)\big| \\
&\quad\le \frac{C(\Omega)}{N_{\text{in}}}\bigg\{\Big|\sum_{p=1}^{N_{\text{in}}} \frac{(\|\nabla u_{\mathfrak{m},1}\|_2^2 - \|\nabla u_{\mathfrak{m},2}\|_2^2)(X_p)}{2}\Big| + \Big|\sum_{p=1}^{N_{\text{in}}} \frac{w(X_p)(u^2_{\mathfrak{m},1} - u^2_{\mathfrak{m},2})(X_p)}{2}\Big| \\
&\qquad + \Big|\sum_{p=1}^{N_{\text{in}}} h(X_p)(u_{\mathfrak{m},1} - u_{\mathfrak{m},2})(X_p)\Big|\bigg\} + \Big|\frac{C(\Omega)}{N_b}\sum_{q=1}^{N_b} g(Y_q)(u_{\mathfrak{m},1} - u_{\mathfrak{m},2})(Y_q)\Big|,
\end{aligned}
$$

where $C(\Omega) = \max\{|\Omega|, |\partial\Omega|\}$. By Lemmas 7 and 8, we have

$$
|\phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \le (W+1)B_{\boldsymbol{\theta}}\,, \qquad |\partial_{x_m}\phi_{\boldsymbol{\theta}}(\boldsymbol{x})| \le W^{L-1}B^L_{\boldsymbol{\theta}}\,.
$$

Also, it holds that

$$
\begin{aligned}
\big|\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) - \phi_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x})\big| &\le 2W^L\sqrt{L}B^{L-1}_{\boldsymbol{\theta}}\big\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\big\|_2\,, \quad \forall \boldsymbol{x}\in\Omega\,, \\
\big|\partial_{x_m}\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) - \partial_{x_m}\phi_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{x})\big| &\le 2W^{2L-1}\sqrt{L}(L+1)B^{2L}_{\boldsymbol{\theta}}\big\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\big\|_2\,, \quad \forall \boldsymbol{x}\in\Omega\,.
\end{aligned}
$$

Then, we have the following estimations. First, it holds that

$$
\begin{aligned}
&\bigg|\frac{1}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}} \frac{\|\nabla u_{\mathfrak{m},1}(X_p)\|_2^2 - \|\nabla u_{\mathfrak{m},2}(X_p)\|_2^2}{2}\bigg| \\
&\quad= \frac{1}{2N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\bigg(\Big\|\sum_{k=1}^{\mathfrak{m}} c_k\nabla\phi^k_{\boldsymbol{\theta},1}(X_p)\Big\|_2^2 - \Big\|\sum_{k=1}^{\mathfrak{m}} c_k\nabla\phi^k_{\boldsymbol{\theta},2}(X_p)\Big\|_2^2\bigg) \\
&\quad\le \frac{1}{2N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\bigg\{\Big|\sum_{m=1}^{d}\Big[\sum_{k=1}^{\mathfrak{m}}|c_k|\Big(\partial_{x_m}\phi^k_{\boldsymbol{\theta},1}(X_p) + \partial_{x_m}\phi^k_{\boldsymbol{\theta},2}(X_p)\Big)\Big] \\
&\qquad\qquad \Big[\sum_{k=1}^{\mathfrak{m}}|c_k|\Big(\partial_{x_m}\phi^k_{\boldsymbol{\theta},1}(X_p) - \partial_{x_m}\phi^k_{\boldsymbol{\theta},2}(X_p)\Big)\Big]\Big|\bigg\} \\
&\quad\le dMW^{L-1}B^L_{\boldsymbol{\theta}}\sum_{k=1}^{\mathfrak{m}}|c_k|\cdot\max_{m,\boldsymbol{x}}\big|\partial_{x_m}\phi^k_{\boldsymbol{\theta},1}(\boldsymbol{x}) - \partial_{x_m}\phi^k_{\boldsymbol{\theta},2}(\boldsymbol{x})\big| \\
&\quad\le dMW^{L-1}B^L_{\boldsymbol{\theta}}\sqrt{\sum_{k=1}^{\mathfrak{m}}|c_k|^2}\sqrt{\sum_{k=1}^{\mathfrak{m}}\max_{m,\boldsymbol{x}}\big|\partial_{x_m}\phi^k_{\boldsymbol{\theta},1}(\boldsymbol{x}) - \partial_{x_m}\phi^k_{\boldsymbol{\theta},2}(\boldsymbol{x})\big|^2} \\
&\quad\le 2dW^{3L-2}\sqrt{L}(L+1)MB^{3L}_{\boldsymbol{\theta}}\|\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}\|_2\big\|\boldsymbol{\theta}^{\mathfrak{m},1}_{\text{in}} - \boldsymbol{\theta}^{\mathfrak{m},2}_{\text{in}}\big\|_2\,,
\end{aligned}
$$

where the fourth step uses Cauchy-Schwarz inequality. Similarly, we have

$$\begin{aligned}
&\left|\frac{1}{2N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}} w(X_p)\big[u^2_{\mathfrak{m},1}(X_p)-u^2_{\mathfrak{m},2}(X_p)\big]\right| \\
&\quad\le B_0M(W+1)B_{\boldsymbol{\theta}}\sqrt{\sum_{k=1}^{\mathfrak{m}}|c_k|^2}\sqrt{\sum_{k=1}^{\mathfrak{m}}\max_{\boldsymbol{x}}\left|\phi^k_{\boldsymbol{\theta},1}(\boldsymbol{x})-\phi^k_{\boldsymbol{\theta},2}(\boldsymbol{x})\right|^2} \\
&\quad\le 2B_0W^L(W+1)\sqrt{L}MB^L_{\boldsymbol{\theta}}\|\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}\|_2\big\|\boldsymbol{\theta}^{\mathfrak{m},1}_{\text{in}}-\boldsymbol{\theta}^{\mathfrak{m},2}_{\text{in}}\big\|_2\,.
\end{aligned}$$

Finally, it holds that

$$\begin{aligned}
&\left|\frac{1}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}\Big[\big(u_{\mathfrak{m},1}(X_p)-u_{\mathfrak{m},2}(X_p)\big)h(X_p)\Big]\right|+\left|\frac{1}{N_b}\sum_{q=1}^{N_b}\Big[\big(u_{\mathfrak{m},1}(Y_q)-u_{\mathfrak{m},2}(Y_q)\big)g(Y_q)\Big]\right| \\
&\quad\le 4B_0W^L\sqrt{L}B^{L-1}_{\boldsymbol{\theta}}\|\boldsymbol{\theta}^{\mathfrak{m}}_{\text{out}}\|_2\big\|\boldsymbol{\theta}^{\mathfrak{m},1}_{\text{in}}-\boldsymbol{\theta}^{\mathfrak{m},2}_{\text{in}}\big\|_2\,.
\end{aligned}$$

Combining above estimations, we complete the proof of this lemma.

C.8 Proof of Lemma 25

In the following, we will focus on the probability of $G^{\mathsf{c}}_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$. The proof will be divided into two steps: (i) the case where $\bar{\mathfrak{m}}=1$ and $R=1$, that is, $G^{\mathsf{c}}_{\mathfrak{m},1,1,\delta}$; (ii) the general case $G^{\mathsf{c}}_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$ where $\mathfrak{m},R\in\mathbb{N}$.

Step 1. For $\bar{\mathfrak{m}}=R=1$, when $\mathfrak{m}=Q$, $G^{\mathsf{c}}_{\mathfrak{m},1,1,\delta}$ denotes the event where, each sub-network weight vector among $(\boldsymbol{\theta}_1)^{[0]}$ through $(\boldsymbol{\theta}_Q)^{[0]}$ satisfies $\|(\boldsymbol{\theta}_\cdot)^{[0]}-\bar{\boldsymbol{\theta}}_1\|_\infty>\delta$. Since $\bar{\boldsymbol{\theta}}_1$ can be treated as a fixed vector in $[-B_{\bar{\boldsymbol{\theta}}},B_{\bar{\boldsymbol{\theta}}}]^{\mathfrak{D}(\bar{W},\bar{L},d)}$, where $\mathfrak{D}(W,L,d)$ is defined in Equation 2, and all sub-network parameters are i.i.d. from $U[-B_{\bar{\boldsymbol{\theta}}},B_{\bar{\boldsymbol{\theta}}}]$, it then holds that

$$\mathbb{P}\Big[\big\|(\boldsymbol{\theta}_i)^{[0]}-\bar{\boldsymbol{\theta}}_1\big\|_\infty\le\delta\Big]\ge\left(\frac{\delta}{2B_{\bar{\boldsymbol{\theta}}}}\right)^{\mathfrak{D}(\bar{W},\bar{L},d)}\ge\left(\frac{\delta}{2B_{\bar{\boldsymbol{\theta}}}}\right)^{\bar{W}(\bar{W}+1)\bar{L}}$$

for any $i\in\{1,\dots,Q\}$. Thus, we have

$$\mathbb{P}\big(G^{\mathsf{c}}_{\mathfrak{m},1,1,\delta}\big)=\mathbb{P}\Big[\forall i\in\{1,\dots,Q\}:\big\|(\boldsymbol{\theta}_i)^{[0]}-\bar{\boldsymbol{\theta}}_1\big\|_\infty>\delta\Big]\le\Big[1-\delta^{\bar{W}(\bar{W}+1)\bar{L}}(2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^Q\,.$$

Step 2. For general $\bar{\mathfrak{m}},R\in\mathbb{N}$, when $\mathfrak{m}=\bar{\mathfrak{m}}\cdot R\cdot Q$, $G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}$ denotes the event where, for each target $\bar{\boldsymbol{\theta}}_k$ in $(\bar{\boldsymbol{\theta}}_1,\dots,\bar{\boldsymbol{\theta}}_{\bar{\mathfrak{m}}})$, at least R weight vectors among $(\boldsymbol{\theta}_k)^{[0]}$ through $(\boldsymbol{\theta}_{\bar{\mathfrak{m}}\cdot R\cdot Q})^{[0]}$ satisfy $\|(\boldsymbol{\theta}_\cdot)^{[0]}-\bar{\boldsymbol{\theta}}_k\|_\infty\le\delta$. It can be readily observed that

$$G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\supseteq\bigcap_{j=1}^{R}\bigcap_{k=1}^{\bar{\mathfrak{m}}}\Big\{\exists i\in\big\{[(j-1)\bar{\mathfrak{m}}+k-1]Q,\dots,[(j-1)\bar{\mathfrak{m}}+k]Q\big\}:\big\|(\boldsymbol{\theta}_i)^{[0]}-\bar{\boldsymbol{\theta}}_k\big\|_\infty\le\delta\Big\}\,.$$

Thus, it holds that

$$G^{\mathsf{c}}_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\subseteq\bigcup_{j=1}^{R}\bigcup_{k=1}^{\bar{\mathfrak{m}}}\Big\{\forall i\in\big\{[(j-1)\bar{\mathfrak{m}}+k-1]Q,\dots,[(j-1)\bar{\mathfrak{m}}+k]Q\big\}:\big\|(\boldsymbol{\theta}_i)^{[0]}-\bar{\boldsymbol{\theta}}_k\big\|_\infty>\delta\Big\}\,.$$

This implies

$$\mathbb{P}\big(G^{\mathsf{c}}_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\big) \le \bar{\mathfrak{m}}\, R\, \mathbb{P}\big(G^{\mathsf{c}}_{\mathfrak{m},1,1,\delta}\big) \le \bar{\mathfrak{m}}\, R\Big[1-\delta^{\bar{W}(\bar{W}+1)\bar{L}}(2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^{Q}.$$

Therefore, we have

$$\mathbb{P}\big(G_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\big) = 1-\mathbb{P}\big(G^{\mathsf{c}}_{\mathfrak{m},\bar{\mathfrak{m}},R,\delta}\big) \ge 1-\bar{\mathfrak{m}}R\Big[1-\delta^{\bar{W}(\bar{W}+1)\bar{L}}(2B_{\bar{\boldsymbol{\theta}}})^{-\bar{W}(\bar{W}+1)\bar{L}}\Big]^{Q}.$$

C.9 Proof of Lemma 26

First, we have

$$\begin{aligned}
&\big|\widehat{\mathcal{L}}(u^*_{\mathfrak{m}}) - \widehat{\mathcal{L}}(u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}})\big| \\
&\quad\le \frac{C(\Omega)}{N_{\mathrm{in}}}\Bigg\{\bigg|\sum_{p=1}^{N_{\mathrm{in}}}\frac{(\|\nabla u^*_{\mathfrak{m}}\|_2^2 - \|\nabla u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\|_2^2)(X_p)}{2}\bigg| + \bigg|\sum_{p=1}^{N_{\mathrm{in}}}\frac{w(X_p)\big[(u^*_{\mathfrak{m}})^2 - u^2_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}\big](X_p)}{2}\bigg| \\
&\qquad + \bigg|\sum_{p=1}^{N_{\mathrm{in}}} h(X_p)(u^*_{\mathfrak{m}} - u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}})(X_p)\bigg|\Bigg\} + \bigg|\frac{C(\Omega)}{N_b}\sum_{q=1}^{N_b} g(Y_q)(u^*_{\mathfrak{m}} - u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}})(Y_q)\bigg|,
\end{aligned}$$

where $C(\Omega) = \max\{|\Omega|, |\partial\Omega|\}$. By Lemmas 7 and 8, we have

$$\begin{aligned}
&\bigg|\frac{1}{N_{\mathrm{in}}}\sum_{p=1}^{N_{\mathrm{in}}}\frac{\|\nabla u^*_{\mathfrak{m}}(X_p)\|_2^2 - \|\nabla u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(X_p)\|_2^2}{2}\bigg| \\
&\quad= \frac{1}{2N_{\mathrm{in}}}\sum_{p=1}^{N_{\mathrm{in}}}\bigg(\Big\|\sum_{k,v}\frac{\bar{c}_k}{R}\cdot\nabla\big(\phi_{\boldsymbol{\theta}}^{s_{k,v}}\big)^{[0]}(X_p)\Big\|_2^2 - \Big\|\sum_{k,v}\frac{\bar{c}_k}{R}\cdot\nabla\phi^k_{\bar{\boldsymbol{\theta}}}(X_p)\Big\|_2^2\bigg) \\
&\quad\le \frac{1}{2N_{\mathrm{in}}}\sum_{p=1}^{N_{\mathrm{in}}}\Bigg\{\bigg|\sum_{m=1}^{d}\Big[\sum_{k,v}\Big|\frac{\bar{c}_k}{R}\Big|\cdot\big[\partial_{x_m}(\phi_{\boldsymbol{\theta}}^{s_{k,v}})^{[0]}(X_p) + \partial_{x_m}\phi^k_{\bar{\boldsymbol{\theta}}}(X_p)\big]\Big]\cdot \\
&\qquad\qquad\Big[\sum_{k,v}\Big|\frac{\bar{c}_k}{R}\Big|\cdot\big(\partial_{x_m}(\phi_{\boldsymbol{\theta}}^{s_{k,v}})^{[0]}(X_p) - \partial_{x_m}\phi^k_{\bar{\boldsymbol{\theta}}}(X_p)\big)\Big]\bigg|\Bigg\} \\
&\quad\le d\bar{M}^2\bar{W}^{L-1}B_{\bar{\boldsymbol{\theta}}}^{\bar{L}}\cdot\max_{\boldsymbol{x},m}\big|\partial_{x_m}(\phi_{\boldsymbol{\theta}}^{s_{k,v}})^{[0]}(\boldsymbol{x}) - \partial_{x_m}\phi^k_{\bar{\boldsymbol{\theta}}}(\boldsymbol{x})\big| \\
&\quad\le d\bar{W}^{3\bar{L}-2}\sqrt{\bar{L}}(\bar{L}+1)\bar{M}^2B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}}\cdot\max_{k,v}\big\|(\boldsymbol{\theta}_{s_{k,v}})^{[0]} - \bar{\boldsymbol{\theta}}_k\big\|_2 \\
&\quad\le 2d\bar{W}^{3\bar{L}-2}\sqrt{\bar{L}}(\bar{L}+1)\bar{M}^2B_{\bar{\boldsymbol{\theta}}}^{3\bar{L}}\sqrt{\bar{W}(\bar{W}+1)\bar{L}}\cdot\max_{k,v}\big\|(\boldsymbol{\theta}_{s_{k,v}})^{[0]} - \bar{\boldsymbol{\theta}}_k\big\|_\infty.
\end{aligned}$$

Similarly, it holds that

$$\begin{aligned}
&\bigg|\frac{1}{N_{\mathrm{in}}}\sum_{p=1}^{N_{\mathrm{in}}}\frac{w(X_p)\big[u^*_{\mathfrak{m}}(X_p)\big]^2 - w(X_p)u^2_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(X_p)}{2}\bigg| \\
&\qquad\le B_0\bar{M}^2(\bar{W}+1)B_{\bar{\boldsymbol{\theta}}}\cdot\max_{\boldsymbol{x}}\big|(\phi_{\boldsymbol{\theta}}^{s_{k,v}})^{[0]}(\boldsymbol{x}) - \phi^k_{\bar{\boldsymbol{\theta}}}(\boldsymbol{x})\big|
\end{aligned}$$

$$\leq 2B_0\bar{W}^{\bar{L}}(\bar{W}+1)\sqrt{\bar{L}}\bar{M}^2B_{\bar{\boldsymbol{\theta}}}^{\bar{L}}\sqrt{\bar{W}(\bar{W}+1)\bar{L}}\cdot\max_{k,v}\left\|(\boldsymbol{\theta}_{s_{k,v}})^{[0]}-\bar{\boldsymbol{\theta}}_k\right\|_\infty.$$

Finally, we have

$$\begin{aligned}&\Big|\frac{1}{N_{\text{in}}}\sum_{p=1}^{N_{\text{in}}}[u_{\mathfrak{m}}^*(X_p)h(X_p)-u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(X_p)h(X_p)]\Big|+\Big|\frac{1}{N_b}\sum_{q=1}^{N_b}[u_{\mathfrak{m}}^*(Y_q)g(Y_q)-u_{\bar{\mathfrak{m}},\bar{\boldsymbol{\theta}}}(Y_q)g(Y_q)]\Big|\\&\quad\leq 4B_0\bar{W}^{\bar{L}}\sqrt{\bar{L}}B_{\bar{\boldsymbol{\theta}}}^{\bar{L}-1}\bar{M}\sqrt{\bar{W}(\bar{W}+1)\bar{L}}\cdot\max_{k,v}\left\|(\boldsymbol{\theta}_{s_{k,v}})^{[0]}-\bar{\boldsymbol{\theta}}_k\right\|_\infty.\end{aligned}$$

Combining above estimations completes the proof.

Appendix D. Complexity of the Projected Gradient Descent Method

In this section, we will compute the algorithmic complexity of one step of the PGD algorithm in Section 2.4. By Duchi et al. (2008) and the explicit ℓ_2-projection formula, the projection step has a complexity of $\mathcal{O}(\dim(\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m}}))$. Thus, only the gradient computation complexity requires attention.

First, we assume that $N_{\text{in}}=N_b=N_s$ in Equation 7, which is consistent with the result in Theorem 1. Thus, Equation 7 can be seen as sum of losses over N_s equally weighted sample pairs $\{X_i,Y_i\}_{i=1}^{N_s}$. When evaluating the complexity of gradient computation, we only need to analyze the loss for a 'representative pair' $\{\boldsymbol{x},\boldsymbol{y}\}$ and then multiply the result by N_s. Then, observe that the gradient update for $\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2$ involves computing the second-order derivatives of the neural network. As this step dominates the overall complexity, we will focus on analyzing this term alone.

Specifically, we will analyze the complexity of calculating

$$\begin{aligned}\frac{\partial\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2}{\partial\boldsymbol{\theta}_{\text{total}}^{\mathfrak{m}}}&=\Big(\frac{\partial\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2}{\partial\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}},\frac{\partial\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2}{\partial\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}}\Big)\\&=:\big(\partial_{\text{in}}\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2,\partial_{\text{out}}\|\nabla u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2\big).\end{aligned}$$

Recall that $u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})=\sum_{k=1}^{\mathfrak{m}}c_k\,\phi_{\boldsymbol{\theta}}^k(\boldsymbol{x})$, $\boldsymbol{\theta}_{\text{out}}^{\mathfrak{m}}=(c_1,c_2,\ldots,c_{\mathfrak{m}})$ and

$$\boldsymbol{\theta}_{\text{in}}^{\mathfrak{m}}=\Big(\underbrace{\boldsymbol{A}_0^1,\boldsymbol{b}_0^1,\ldots,\boldsymbol{A}_{L-1}^1,\boldsymbol{b}_{L-1}^1}_{\phi_{\boldsymbol{\theta}}^1(\boldsymbol{x})},\ldots\ldots,\underbrace{\boldsymbol{A}_0^{\mathfrak{m}},\boldsymbol{b}_0^{\mathfrak{m}},\ldots,\boldsymbol{A}_{L-1}^{\mathfrak{m}},\boldsymbol{b}_{L-1}^{\mathfrak{m}}}_{\phi_{\boldsymbol{\theta}}^{\mathfrak{m}}(\boldsymbol{x})}}\Big)$$

For simplicity, we will abbreviate $u_{\mathfrak{m},\boldsymbol{\theta}}(\boldsymbol{x})$ as u, $\phi_{\boldsymbol{\theta}}^k(\boldsymbol{x})$ as ϕ_k. Additionally, we assume that the input dimension $d=W$ and all $\boldsymbol{A}_\ell^k\in\mathbb{R}^{W\times W},0\leq\ell<L-1$. In this way, all subnetworks share a consistent parameter structure. Direct calculation yields

$$\partial_{\text{out}}\|\nabla u\|_2^2=\big(\nabla\phi_1^\top(2\nabla u),\ldots,\nabla\phi_{\mathfrak{m}}^\top(2\nabla u)\big),\quad\nabla u=c_1\nabla\phi_1+\cdots+c_{\mathfrak{m}}\nabla\phi_{\mathfrak{m}}.\tag{49}$$

Therefore, once the complexity of $\nabla\phi_k$ is determined, the complexity of $\partial_{\text{out}}\|\nabla u\|_2^2$ will be clear. Before diving into the computations, we need to outline the matrix differentiation rules relevant to this section.

In summary, our focus is the derivative of a scalar-valued matrix function $f(\boldsymbol{X}) \in \mathbb{R}$ with respect to $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, defined as $\partial f/\partial \boldsymbol{X} := (\partial f/\partial \boldsymbol{X}_{ij})$. Its computation relies on the following key relationship:

$$\mathrm{d}f = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\partial f}{\partial \boldsymbol{X}_{ij}}\mathrm{d}\boldsymbol{X}_{ij} = \mathrm{tr}\left(\frac{\partial f}{\partial \boldsymbol{X}}^{\top}\mathrm{d}\boldsymbol{X}\right), \quad \mathrm{d}\boldsymbol{X} := (\mathrm{d}\boldsymbol{X}_{ij}).$$

That is to say, the computation of $\partial f/\partial \boldsymbol{X}$ involves two steps: (1) deriving the relationship between the differentials $\mathrm{d}f$ and $\mathrm{d}\boldsymbol{X}$ based on the specific form of $f(X)$; and (2) leveraging the properties of the trace operator to express $\mathrm{d}f$ as $\mathrm{tr}(\boldsymbol{A}^{\top}\mathrm{d}\boldsymbol{X})$, where $\boldsymbol{A}$ represents the sought-after matrix derivative. For the former step, we need the following rules:

$$\begin{aligned}
&\mathrm{d}(\boldsymbol{X}\pm\boldsymbol{Y}) = \mathrm{d}\boldsymbol{X}\pm\mathrm{d}\boldsymbol{Y}; \quad \mathrm{d}(\boldsymbol{X}\boldsymbol{Y}) = \boldsymbol{X}\mathrm{d}\boldsymbol{Y} + (\mathrm{d}\boldsymbol{X})\boldsymbol{Y}; \quad \mathrm{d}(\boldsymbol{X}^{\top}) = (\mathrm{d}\boldsymbol{X})^{\top};\\
&\mathrm{d}\rho(\boldsymbol{X}) = \rho'(\boldsymbol{X})\odot\mathrm{d}\boldsymbol{X}; \quad \mathrm{d}(\boldsymbol{X}\odot\boldsymbol{Y}) = \boldsymbol{X}\odot\mathrm{d}\boldsymbol{Y} + \mathrm{d}\boldsymbol{X}\odot\boldsymbol{Y},
\end{aligned}$$

where ρ denotes an element-wise function, and $\odot$ represents the element-wise Hadamard product. For the latter step, the following rules are necessary:

$$\begin{aligned}
&\mathrm{tr}(\boldsymbol{A}\pm\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}(\boldsymbol{B}); \quad \mathrm{tr}\big(\boldsymbol{A}^{\top}\boldsymbol{B}\big) = \mathrm{tr}\big(\boldsymbol{B}\boldsymbol{A}^{\top}\big)\\
&\mathrm{tr}(\boldsymbol{A}^{\top}) = \mathrm{tr}(\boldsymbol{A}); \quad \mathrm{tr}\big(\boldsymbol{A}^{\top}(\boldsymbol{B}\odot\boldsymbol{C})\big) = \mathrm{tr}\big((\boldsymbol{A}\odot\boldsymbol{B})^{\top}\boldsymbol{C}\big),
\end{aligned}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ have the same shape. With these, we present the computation of $\nabla\phi_k$, which is known as the 'backpropagation algorithm' in deep learning.

The process starts with the forward pass that ultimately reaches ϕ_k, during which the following intermediate quantities will be stored:

$$\begin{aligned}
&\boldsymbol{z}_0 = \boldsymbol{A}_0\boldsymbol{x} + \boldsymbol{b}_0; \quad \boldsymbol{x}_1 = \rho(\boldsymbol{z}_0); \quad \boldsymbol{z}_1 = \boldsymbol{A}_1\boldsymbol{x}_1 + \boldsymbol{b}_1; \quad \boldsymbol{x}_2 = \rho(\boldsymbol{z}_1); \quad \cdots\\
&\boldsymbol{z}_{L-2} = \boldsymbol{A}_{L-2}\boldsymbol{x}_{L-2} + \boldsymbol{b}_{L-2}; \quad \boldsymbol{x}_{L-1} = \rho(\boldsymbol{z}_{L-2}); \quad \phi_k = \boldsymbol{A}_{L-1}\boldsymbol{x}_{L-1} + \boldsymbol{b}_{L-1},
\end{aligned}$$

where the parameter matrices and biases omit the subscript k for simplicity. Note that for $0 \le i < L-2$, the floating-point operations for $\boldsymbol{z}_i$ are $2W^2$, and the operations for $\boldsymbol{x}_{i+1}$ are W. Therefore, the floating-point operations for ϕ_k are $W(2W+1)(L-1)+2W$.

Next, the backward computation of $\nabla\phi_k$ begins, storing the following intermediate quantities, which are required for evaluating the second-order parameter derivatives in $\partial_{\mathrm{in}}\|\nabla u\|_2^2$.

$$\begin{aligned}
&\boldsymbol{s}_0 = \rho'(\boldsymbol{z}_{L-2}); \quad \boldsymbol{q}_0 = \boldsymbol{A}_{L-1}^{\top}\odot\boldsymbol{s}_0; \quad \boldsymbol{p}_1 = \boldsymbol{A}_{L-2}^{\top}\boldsymbol{q}_0;\\
&\boldsymbol{s}_1 = \rho'(\boldsymbol{z}_{L-3}); \quad \boldsymbol{q}_1 = \boldsymbol{p}_1^{\top}\odot\boldsymbol{s}_1; \quad \boldsymbol{p}_2 = \boldsymbol{A}_{L-3}^{\top}\boldsymbol{q}_0; \quad \cdots\\
&\boldsymbol{s}_{L-2} = \rho'(\boldsymbol{z}_0); \quad \boldsymbol{q}_{L-2} = \boldsymbol{p}_{L-2}\odot\boldsymbol{s}_{L-2}; \quad \nabla\phi_k = \boldsymbol{A}_0^{\top}\boldsymbol{q}_{L-2}.
\end{aligned}$$

The additional floating-point operations are $W(2W+1)(L-1)$, leveraging the intermediate quantities stored during the forward pass. Thus, the total floating-point operations for computing a single $\nabla\phi_k$ are $2W(2W+1)(L-1)+2W$, and finally, by Equation 49, the complexity of $\partial_{\mathrm{out}}\|\nabla u\|_2^2$ is $2\mathfrak{m}W(2W+1)(L-1)+6\mathfrak{m}W-W$, during which $2\nabla u$ is stored.

Now, let's analyze $\partial_{\mathrm{in}}\|\nabla u\|_2^2$. First, note that

$$\mathrm{d}\|\nabla u\|_2^2 = (2\nabla u)^{\top}(\mathrm{d}\nabla u) = (2\nabla u)^{\top}\big[c_1(\mathrm{d}\nabla\phi_1) + \cdots + (\mathrm{d}\nabla\phi_{\mathfrak{m}})\big]$$

$$= (2c_1\nabla u)^\top(\mathrm{d}\nabla\phi_1) + \cdots + (2c_{\mathfrak{m}}\nabla u)^\top(\mathrm{d}\nabla\phi_{\mathfrak{m}})\,.$$

It is evident that the partial derivatives with respect to the parameters of any given ϕ_k are solely associated with $(2c_k\nabla u)^\top(\mathrm{d}\nabla\phi_k)$. Given the identical subnetworks, it suffices to determine the complexity of the partial derivatives in a single $(2c_k\nabla u)^\top(\mathrm{d}\nabla\phi_k)$, which, multiplied by $\mathfrak{m}$, gives the total complexity of $\partial_{\mathrm{in}}\|\nabla u\|_2^2$.

We first define some auxiliary quantities. Let $\boldsymbol{\pi}_0 = 2c_k\nabla u$. For $0 \le i \le L-2$, let

$$\begin{aligned}
&\boldsymbol{\chi}_i = \boldsymbol{q}_{L-2-i}\,\boldsymbol{\pi}_i^\top\,; \quad \boldsymbol{\iota}_i = \boldsymbol{A}_i\,\boldsymbol{\pi}_i\,; \quad \boldsymbol{\pi}_{i+1} = \boldsymbol{\iota}_i \odot \boldsymbol{s}_{L-2-i}\,;\\
&\boldsymbol{r}_i^0 = \boldsymbol{\iota}_i \odot \boldsymbol{p}_{L-2-i} \odot \rho''(\boldsymbol{z}_i)\,; \quad \boldsymbol{r}_i^j = \boldsymbol{A}_{i-j+1}^\top \boldsymbol{r}_i^{j-1} \odot \rho''(\boldsymbol{z}_{i-j})\,,\ 0 < j \le i\,,
\end{aligned}$$

where the subscript k is also omitted for simplicity. Using all previously stored intermediate quantities, the incremental floating-point operation counts for the above five quantities are as follows: W^2, $W(2W-1)$, W, $3W$ and $2W^2$. Denoting by $\boldsymbol{\Upsilon}_i := \partial_{\boldsymbol{A}_i}\|\nabla u\|_2^2$ and $\boldsymbol{\upsilon}_j := \partial_{\boldsymbol{b}_j}\|\nabla u\|_2^2$, we then have

$$(2c_k\nabla u)^\top(\mathrm{d}\nabla\phi_k) = \sum_{i=0}^{L-1}\mathrm{tr}(\boldsymbol{\Upsilon}_i^\top\,\mathrm{d}\boldsymbol{A}_i) + \sum_{j=0}^{L-2}\boldsymbol{\upsilon}_j^\top\,\mathrm{d}\boldsymbol{b}_j\,,$$

where $\boldsymbol{\Upsilon}_{L-1} = \boldsymbol{\pi}_{L-1}^\top$, and for $0 \le k \le L-2$,

$$\boldsymbol{b}_k = \sum_{l=k}^{L-2}\boldsymbol{r}_l^{l-k}\,, \quad \boldsymbol{\Upsilon}_k = \boldsymbol{\chi}_k + \sum_{l=k}^{L-2}\boldsymbol{r}_l^{l-k}\boldsymbol{x}_k^\top\,.$$

Thus, the complexity of the ϕ_k component in $\partial_{\mathrm{in}}\|\nabla u\|_2^2$ is $(L-1)(L-2)(2W^2+W) + (L-1)(5W^2+3W)$. Based on all analyses, the complexity of the gradient update is $\mathcal{O}(\mathfrak{m}W^2L^2N_s)$. Since $\dim(\boldsymbol{\theta}_{\mathrm{total}}^{\mathfrak{m}}) = \mathcal{O}(\mathfrak{m}W^2L)$, the complexity of projection is $\mathcal{O}(\mathfrak{m}W^2L)$. Finally, we deduce that the complexity of one step of projected gradient descent in Section 2.4 is $\mathcal{O}(\mathfrak{m}W^2L^2N_s)$.

References

Shmuel Agmon, Avron Douglis, and Louis Nirenberg. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions. i. *Communications on Pure and Applied Mathematics*, 12(4):623–727, 1959.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

Cosmin Anitescu, Elena Atroshchenko, Naif Alajlan, and Timon Rabczuk. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials and Continua*, 59(1):345–359, 2019.

Francis Bach. Learning theory from first principles, 2023.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(1):2285–2301, 2019.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta Numerica*, 30:87–201, 2021.

Benedikt Bauer and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4):2261–2285, 2019.

Christian Beck, Arnulf Jentzen, and Benno Kuckuck. Full error analysis for the training of deep neural networks. *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, 25(02):2150020, 2022.

Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. Does data interpolation contradict statistical optimality? In *International Conference on Artificial Intelligence and Statistics*, pages 1611–1619. PMLR, 2019b.

Julius Berner, Markus Dablander, and Philipp Grohs. Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 16615–16627. Curran Associates, Inc., 2020.

Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *Mathematical Aspects of Deep Learning*, page 1, 2022.

Susanne Brenner and Ridgway Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007.

Mo Chen, Zhao Ding, Yuling Jiao, Xiliang Lu, Peiying Wu, and Jerry Zhijian Yang. Convergence analysis of PINNs with over-parameterization. *Communications in Computational Physics*, 37(4):942–974, 2025.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Philippe G Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2002.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

Yongcheng Dai, Bangti Jin, Ramesh Sau, and Zhi Zhou. Solving elliptic optimal control problems via neural networks and optimality system. *arXiv preprint arXiv:2308.11925*, 2023.

Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021.

Zhao Ding, Yuling Jiao, Xiliang Lu, Peiying Wu, and Jerry Zhijian Yang. Convergence analysis of deep Ritz method with over-parameterization. *Neural Networks*, 184:107110, 2025.

Selina Drews and Michael Kohler. Analysis of the expected L_2 error of an over-parametrized deep neural network estimate learned by gradient descent without regularization. *arXiv preprint arXiv:2311.14609*, 2023.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

Chenguang Duan, Yuling Jiao, Yanming Lai, Dingwei Li, Jerry Zhijian Yang, et al. Convergence rate analysis for deep Ritz method. *Communications in Computational Physics*, 31(4):1020–1048, 2022.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the L_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, pages 272–279, 2008.

Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

Evarist Giné and Richard Nickl. *Mathematical foundations of infinite-dimensional statistical models*. Cambridge University Press, 2021.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference on Learning Theory*, pages 297–299. PMLR, 2018.

Philipp Grohs and Gitta Kutyniok. *Mathematical aspects of deep learning*. Cambridge University Press, 2022.

Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

Ingo Gühring, Gitta Kutyniok, and Philipp Petersen. Error bounds for approximations with deep ReLU neural networks in $W_{s,p}$ norms. *Analysis and Applications*, 18(05):803–859, 2020.

Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2): 949–986, 2022.

Qingguo Hong, Jonathan W Siegel, and Jinchao Xu. Rademacher complexity and numerical quadrature analysis of stable neural networks with applications to numerical PDEs. *arXiv preprint arXiv:2104.02903*, 2021.

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

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Tianhao Hu, Bangti Jin, and Zhi Zhou. Solving elliptic problems with singular sources using singularity splitting deep Ritz method. *SIAM Journal on Scientific Computing*, 45 (4):A2043–A2074, 2023.

Tianhao Hu, Bangti Jin, and Zhi Zhou. Solving Poisson problems in polygonal domains with singularity enriched physics informed neural networks. *SIAM Journal on Scientific Computing*, 46(4):C369–C398, 2024.

Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, Tuan Anh Nguyen, and Philippe von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *Proceedings of the Royal Society A*, 476(2244):20190630, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Arnulf Jentzen and Timo Welti. Overall error analysis for the training of deep neural networks via stochastic gradient descent with random initialisation. *Applied Mathematics and Computation*, 455:127907, 2023.

Xia Ji, Yuling Jiao, Xiliang Lu, Pengcheng Song, and Fengru Wang. Deep Ritz method for elliptical multiple eigenvalue problems. *Journal of Scientific Computing*, 98(2):48, 2024.

Yuling Jiao, Yanming Lai, Dingwei Li, Xiliang Lu, Fengru Wang, Jerry Zhijian Yang, et al. A rate of convergence of physics informed neural networks for the linear second order elliptic PDEs. *Communications in Computational Physics*, 31(4):1272–1295, 2022.

Yuling Jiao, Yanming Lai, Yisu Lo, Yang Wang, and Yunfei Yang. Error analysis of deep Ritz methods for elliptic equations. *Analysis and Applications*, 2023a.

Yuling Jiao, Yanming Lai, Xiliang Lu, Fengru Wang, Jerry Zhijian Yang, and Yuanyuan Yang. Deep neural networks with ReLU-sine-exponential activations break curse of dimensionality in approximation on Hölder class. *SIAM Journal on Mathematical Analysis*, 55(4):3635–3649, 2023b.

Yuling Jiao, Xiliang Lu, Jerry Zhijian Yang, Cheng Yuan, and Pingwen Zhang. Improved analysis of PINNs: Alleviate the CoD for compositional solutions. *Annals of Applied Mathematics*, 39:239–263, 2023c.

Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, 2023d.

Yuling Jiao, Yang Wang, and Yunfei Yang. Approximation bounds for norm constrained neural networks with applications to regression and GANs. *Applied and Computational Harmonic Analysis*, 65:249–278, 2023e.

Yuling Jiao, Jerry Zhijian Yang, Junyu Zhou, et al. A rate of convergence of weak adversarial neural networks for the second order parabolic PDEs. *Communications in Computational Physics*, 34(3):813–836, 2023f.

Yuling Jiao, Xiliang Lu, Peiying Wu, and Jerry Zhijian Yang. Convergence analysis for over-parameterized deep learning. *Communications in Computational Physics*, 36(1):71–103, 2024.

Yuling Jiao, Yanming Lai, and Yang Wang. Error analysis of three-layer neural network trained with PGD for deep Ritz method. *IEEE Transactions on Information Theory*, 2025. In press.

Varun Kanade, Patrick Rebeschini, and Tomas Vaskevicius. Exponential tail local rademacher complexity risk bounds without the bernstein condition. *Journal of Machine Learning Research*, 25(388):1–43, 2024.

Michael Kohler and Adam Krzyzak. Over-parametrized deep neural networks minimizing the empirical risk do not generalize well. *Bernoulli*, 27(4):2564–2597, 2021.

Michael Kohler and Adam Krzyzak. On the rate of convergence of an over-parametrized deep neural network regression estimate with ReLU activation function learned by gradient descent. preprint, 2023a. URL `https://www2.mathematik.tu-darmstadt.de/~kohler/preprint23_01.pdf`.

Michael Kohler and Adam Krzyzak. On the rate of convergence of an over-parametrized transformer classifier learned by gradient descent. *arXiv preprint arXiv:2312.17007*, 2023b.

Michael Kohler and Sophie Langer. On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics*, 49(4):2231–2249, 2021.

Vladimir Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

Gitta Kutyniok, Philipp Petersen, Mones Raslan, and Reinhold Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *Constructive Approximation*, 55 (1):73–125, 2022.

Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for DeepOnets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 48(3):1329–1347, 2020.

Yulei Liao and Pingbing Ming. Deep Nitsche method: Deep Ritz method with essential boundary conditions. *Communications in Computational Physics*, 29(5):1365–1384, 2021.

Yulei Liao and Pingbing Ming. Spectral Barron space and deep neural network approximation. *arXiv preprint arXiv:2309.00788*, 2023.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 2022.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021a.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021b.

Yiping Lu, Chao Ma, Yulong Lu, Jianfeng Lu, and Lexing Ying. A mean field analysis of deep ResNet and beyond: Towards provably optimization via overparameterization from depth. In *International Conference on Machine Learning*, pages 6426–6436. PMLR, 2020.

Yiping Lu, Haoxuan Chen, Jianfeng Lu, Lexing Ying, and Jose Blanchet. Machine learning for elliptic PDEs: Fast rate generalization bound, neural scaling law and minimax optimality. In *International Conference on Learning Representations*, 2021c.

Yulong Lu, Jianfeng Lu, and Min Wang. A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations. In *Conference on Learning Theory*, pages 3196–3241. PMLR, 2021d.

Chao Ma, Lei Wu, et al. The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022.

Arvind Mahankali, Haochen Zhang, Kefan Dong, Margalit Glasgow, and Tengyu Ma. Beyond NTK with vanilla gradient descent: A mean-field analysis of neural networks with polynomial width, samples, and time. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

Shahar Mendelson. Learning without concentration for general loss functions. *Probability Theory and Related Fields*, 171(1):459–502, 2018.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. *IMA Journal of Numerical Analysis*, 42(2):981–1022, 2022.

Siddhartha Mishra and T Konstantin Rusch. Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences. *SIAM Journal on Numerical Analysis*, 59(3):1811–1834, 2021.

Johannes Müller and Guido Montúfar. Geometry and convergence of natural policy gradient methods. *Information Geometry*, 7(Suppl 1):485–523, 2024.

Johannes Müller and Marius Zeinhofer. Error estimates for the variational training of neural networks with boundary penalty. *arXiv preprint arXiv:2103.01007*, 2021.

Johannes Müller and Marius Zeinhofer. Achieving high accuracy with PINNs via energy natural gradient descent. In *International Conference on Machine Learning*, volume 202, pages 25471–25485. PMLR, 23–29 Jul 2023.

Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21: 174–1, 2020.

Preetum Nakkiran, Prayaag Venkat, Sham M Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. In *International Conference on Learning Representations*, 2020.

Quynh Nguyen. On the proof of global convergence of gradient descent for deep ReLU networks with linear widths. In *International Conference on Machine Learning*, pages 8056–8062. PMLR, 2021.

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

Philipp Christian Petersen. Neural network theory. *University of Vienna*, 2020.

Allan Pinkus. Approximation theory of the MLP model. *Acta Numerica*, 8:143–195, 1999.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Johannes Schmidt-Hieber et al. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28(5):1768–1811, 2020.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network with approximation error being reciprocal of width to power of square root of depth. *Neural Computation*, 33(4): 1005–1036, 2021a.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Neural network approximation: Three hidden layers are enough. *Neural Networks*, 141:160–173, 2021b.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Optimal approximation rate of ReLU networks in terms of width and depth. *Journal de Mathématiques Pures et Appliquées*, 157:101–135, 2022.

Yeonjong Shin. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. *Communications in Computational Physics*, 28 (5):2042–2074, 2020.

Jonathan W Siegel and Jinchao Xu. Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321, 2020.

Justin A. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

Hwijae Son, Jin Woo Jang, Woo Jin Han, and Hyung Ju Hwang. Sobolev training for the neural network solutions of PDEs. *arXiv preprint arXiv:2101.08932*, 2021.

Elias M Stein. *Singular integrals and differentiability properties of functions*. Princeton university press, 1970.

Taiji Suzuki. 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*, 2018.

Taiji Suzuki and Atsushi Nitanda. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Matus Telgarsky. Deep learning theory lecture notes, 2021.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

Sara A. van de Geer. *Empirical processes in M-estimation*, volume 6. Cambridge University Press, 2000.

Aad Van Der Vaart and Jon Wellner. *Weak convergence*. Springer, 1996.

Gal Vardi, Gilad Yehudai, and Ohad Shamir. On the optimal memorization power of ReLU neural networks. In *International Conference on Learning Representations*, 2021.

Roman Vershynin. Memory capacity of neural networks with threshold and rectified linear unit activations. *SIAM Journal on Mathematics of Data Science*, 2(4):1004–1033, 2020.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

E Weinan. Machine learning and computational mathematics. *Communications in Computational Physics*, 28(5):1639–1670, 2020.

E. Weinan and Ting Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1): 1–12, 2017.

E Weinan, Chao Ma, and Lei Wu. A priori estimates of the population risk for two-layer neural networks. *Communications in Mathematical Sciences*, 17(5):1407–1425, 2019.

E Weinan, Chao Ma, and Lei Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics*, pages 1–24, 2020.

Yunbei Xu and Assaf Zeevi. Towards optimal problem dependent generalization error bounds in statistical learning theory. In *Advances in Neural Information Processing Systems*, 2021.

Yahong Yang and Juncai He. Deeper or wider: A perspective from optimal generalization error with sobolev loss. In *International Conference on Machine Learning*, pages 56109–56138. PMLR, 2024.

Yunfei Yang and Ding-Xuan Zhou. Optimal rates of approximation by shallow $ReLU^k$ neural networks and applications to nonparametric regression. *Constructive Approximation*, pages 1–32, 2024.

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

Dmitry Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In *Conference on Learning Theory*, pages 639–649. PMLR, 2018.

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

Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 13005–13015, 2020.

Hao Yu, Yixiao Guo, and Pingbing Ming. Generalization error estimates of machine learning methods for solving high dimensional Schrödinger eigenvalue problems. *arXiv preprint arXiv:2408.13511*, 2024.

Yaohua Zang, Gang Bao, Xiaojing Ye, and Haomin Zhou. Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, 411: 109409, 2020.

Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis. Time-Frequency and Time-Scale Analysis, Wavelets, Numerical Algorithms, and Applications*, 48(2):787–794, 2020.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.