Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2025-0102

Vol. **40**, No. 1, pp. 27-60
July 2026

A Complete Error Analysis of PINNs for Elliptic Equations Using Projected Stochastic Gradient Descent

Yuling Jiao[2,3,5], Ruoxuan Li[1], Defeng Sun[6], Peiying Wu[1] and Jerry Zhijian Yang[1,2,3,4,*]

[1] *School of Mathematics and Statistics, Wuhan University, Wuhan 430072, P.R. China.*
[2] *National Center for Applied Mathematics in Hubei, Wuhan University, Wuhan 430072, P.R. China.*
[3] *Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan 430072, P.R. China.*
[4] *Institute for Math & AI, Wuhan University, Wuhan 430072, P.R. China.*
[5] *School of Artificial Intelligence, Wuhan University, Wuhan 430072, P.R. China.*
[6] *Department of Applied Mathematics and Research Center for Intelligent Operations Research, The Hong Kong Polytechnic University, Hong Kong, P.R. China.*

Received 19 April 2025; Accepted (in revised version) 29 August 2025

Abstract. Physics-informed neural networks (PINNs) have recently gained attention as a powerful and efficient tool for solving partial differential equations (PDEs). Despite their empirical success, the theoretical understanding of PINNs, especially in the context of over-parameterization, remains incomplete. This paper presents a complete error analysis of over-parameterized PINNs for elliptic equations using projected stochastic gradient descent (PSGD) optimization. Our analysis rigorously examines the interplay of approximation error, statistical error, and optimization error, offering a unified framework for understanding the convergence behavior of PINNs. By leveraging the properties of PSGD, we establish convergence rates and derive conditions on neural network architecture, training sample requirements, and optimization parameters to ensure specified accuracy.

AMS subject classifications: 65M15, 65N15, 65Y20

Key words: Physics-informed neural networks, projected stochastic gradient descent, complete error analysis, over-parameterization.

*Corresponding author. *Email addresses:* yulingjiaomath@whu.edu.cn (Y. Jiao), ruoxuanli.math@whu.edu.cn (R. Li), defeng.sun@polyu.edu.hk (D. Sun), peiyingwu@whu.edu.cn (P. Wu), zjyang.math@whu.edu.cn (J. Z. Yang)

1 Introduction

Traditional numerical methods, such as the finite element method [9, 12], have proven to be very effective for solving low-dimensional PDEs. However, they encounter significant difficulties when applied to high-dimensional problems. The impressive success of deep learning in handling high-dimensional data has paved the way for employing deep neural networks in solving high-dimensional PDEs [1, 8, 16, 17, 19, 31, 35, 45, 47, 52]. Due to the excellent approximation power of deep neural networks, several numerical schemes have been proposed for solving PDEs, including the deep Ritz method [16], PDE-net [34], PINNs [45] and weak adversarial networks [52]. Among the various techniques developed, PINNs have emerged as a particularly powerful approach [45]. PINNs not only leverage the robust approximation abilities of deep learning but also seamlessly incorporate the underlying physical laws of the PDEs, making them highly effective for solving high-dimensional problems [25, 43, 44]. The success of PINNs has spurred deeper theoretical analysis, highlighting the need for comprehensive error analysis in deep learning, including approximation, generalization, and optimization errors.

While several studies have investigated the theoretical mechanisms of PINNs [13, 21–23, 26, 32, 36, 38, 39, 41, 46, 49, 50], these analyses exhibit two key limitations. First, they are typically conducted in scenarios where the number of neural network parameters is smaller than the number of training samples. Second, these analyses often do not address optimization errors, which are crucial for a comprehensive understanding of model performance. Specifically, over-parameterized deep neural networks, where the number of parameters significantly exceeds the sample size, are frequently employed in real-world applications due to their computational efficiency during training. Although extensive research has examined the role of over-parameterization in linear and kernel models, particularly in relation to the double descent phenomenon [2–7, 20, 33, 42], the underlying reasons for the effectiveness of over-parameterized deep neural networks remain unclear. Providing theoretical guarantees in such regimes continues to be a fundamental yet challenging problem. Recent studies have reported convergence results for over-parameterized norm-controlled neural networks in both regression and PDE settings [11, 29, 30, 51]. However, these analyses typically assume that optimization algorithms such as gradient descent (GD) or stochastic gradient descent (SGD) yield the empirical risk minimization (ERM) estimator, the theoretically optimal solution, thus neglecting the influence of optimization errors introduced during the training process.

In this work, we establish a complete error analysis of PINNs for elliptic equations in the over-parameterized setting using projected stochastic gradient descent (PSGD) optimization. Our analysis accounts for all three key error components: approximation error, statistical (generalization) error, and optimization error. By integrating these error components within a unified theoretical framework, we derive explicit convergence rates and precise conditions for achieving specified accuracy levels when solving elliptic boundary value problems. This represents a significant advancement in the theoretical understanding of PINNs.

This paper is structured as follows. Section 2 introduces the problem setup, describes the neural network structure, and outlines the PSGD algorithm for optimization. Section 3 presents a comprehensive end-to-end error analysis for solving second-order elliptic equations with PINNs. Our main result, presented in Theorem 3.5, provides a complete characterization of the convergence behavior under suitable choices of network architecture and optimization parameters.

1.1 Notation

We establish notation for this paper as follows. Bold-faced letters denote vectors, while capital letters represent matrices or fixed parameters. The symbol C denotes an absolute constant, whereas $C(a,b)$ or $C_i(a,b)$ represent constants that depend only on parameters a and b. For positive functions $f(x)$ and $g(x)$, we write $f(x)=\mathcal{O}(g(x))$ when $f(x)\leq Cg(x)$ for some $C>0$, and use $\widetilde{\mathcal{O}}(\cdot)$ to omit logarithmic factors.

Let $\mathbb{N}$ denote the natural numbers and $\mathbb{N}^+:=\{x\in\mathbb{N}\,|\,x>0\}$ the positive integers. For $x\in\mathbb{R}$, $\lfloor x\rfloor:=\max\{k\in\mathbb{N}:k\leq x\}$ and $\lceil x\rceil:=\min\{k\in\mathbb{N}:k\geq x\}$ denote the floor and ceiling functions. For $N\in\mathbb{N}^+$, $[N]:=\{1,2,\cdots,N\}$ represents the set of integers from 1 to N. Given a vector $\boldsymbol{x}=(x_1,\cdots,x_d)^{\mathrm{T}}\in\mathbb{R}^d$, its ℓ^2-norm and ℓ^∞-norm are defined as $\|\boldsymbol{x}\|_2:=\sqrt{\sum_{i=1}^d x_i^2}$ and $\|\boldsymbol{x}\|_\infty:=\max_{1\leq i\leq d}|x_i|$. For a multi-index $\boldsymbol{\alpha}\in\mathbb{N}^d$, we define $\|\boldsymbol{\alpha}\|_1:=\alpha_1+\cdots+\alpha_d$ and $\boldsymbol{\alpha}!:=\alpha_1!\cdot\cdots\cdot\alpha_d!$.

For an open set $\mathcal{D}\subset\mathbb{R}^d$ and a function $f:\mathcal{D}\to\mathbb{R}$, we define the $L^p(\mathcal{D})$ norm as:

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

The derivative of order $\boldsymbol{\alpha}$ of f is denoted by:

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

We denote by $C^s(\mathcal{D})$ the set of s-times continuously differentiable functions on $\mathcal{D}$ for $s\in\mathbb{N}\cup\{\infty\}$. When $\bar{\mathcal{D}}$ is compact and $f\in C^s(\mathcal{D})$, we define:

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

For $s\in\mathbb{N}$ and $1\leq p<\infty$, the Sobolev space $W^{s,p}(\mathcal{D})$ is defined as:

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

When $p=2$, we denote $H^s(\mathcal{D}):=W^{s,2}(\mathcal{D})$ for any $s\in\mathbb{N}$. For $f\in W^{s,p}(\mathcal{D})$ with $1\leq p<\infty$, the Sobolev norm is given by:

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

For $p=\infty$, we define:

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

For $f\in L^2(\mathcal{D})$, we define:

$$\|f\|_{H^{1/2}(\mathcal{D})} := \left(\|f\|^2_{L^2(\mathcal{D})} + \int_{\mathcal{D}}\int_{\mathcal{D}} \frac{|f(\boldsymbol{x})-f(\boldsymbol{y})|^2}{\|\boldsymbol{x}-\boldsymbol{y}\|^{d+1}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}\right)^{1/2}.$$

2 Preliminaries

In this section, we establish the mathematical foundation for our analysis. We begin with the problem setup by introducing the PINNs formulation for elliptic equations, followed by the neural network architecture, and conclude with the projected stochastic gradient descent algorithm used for optimization.

2.1 Problem set-up

We first recall the PINNs proposed in [45]. Within the d-dimensional unit cube $[0,1]^d$, let $\mathcal{D}$ be a bounded convex domain with boundary $\partial\mathcal{D}$. Consider the following linear second-order elliptic equation with Dirichlet boundary conditions:

$$\begin{cases} -\sum_{i,j=1}^{d} \kappa_{ij}\partial_{s_i}\partial_{s_j} v + \sum_{i=1}^{d} \nu_i \partial_{s_i} v + \mu v = f & \text{in } \mathcal{D}, \\ v = \psi & \text{on } \partial\mathcal{D}, \end{cases} \tag{2.1}$$

where $\kappa_{ij}\in C(\bar{\mathcal{D}})$, $\nu_i,\mu\in L^\infty(\mathcal{D})$, $f\in L^2(\mathcal{D})$, $\psi\in L^2(\partial\mathcal{D})$ with the strictly elliptic condition, i.e., there exists a constant $\tau_0>0$ such that $\sum_{i,j}\kappa_{ij}\xi_i\xi_j\ge\tau_0|\boldsymbol{\xi}|^2$, $\forall \boldsymbol{s}\in\mathcal{D},\boldsymbol{\xi}\in\mathbb{R}^d$. Assume that (2.1) has a unique strong solution $v_*\in C^m(\bar{\mathcal{D}})$ with $m\ge 3$. Define the coefficient norms $R_\kappa=\max_{i,j}\{\|\kappa_{ij}\|_{C(\bar{\mathcal{D}})}\}$ and $R_\nu=\max_i\{\|\nu_i\|_{L^\infty(\mathcal{D})}\}$. Further, let $R_\mu=\|\mu\|_{L^\infty(\mathcal{D})}$, $R_f=\|f\|_{L^2(\mathcal{D})}$, and $R_\psi=\|\psi\|_{L^2(\partial\mathcal{D})}$. We set $R_0=\max\{R_\kappa,R_\nu,R_\mu,R_f,R_\psi\}$. The residual of (2.1) is defined as

$$\mathcal{J}(v) := \int_{\mathcal{D}} \Big(-\sum_{i,j=1}^{d}\kappa_{ij}\partial_{s_i}\partial_{s_j}v + \sum_{i=1}^{d}\nu_i\partial_{s_i}v + \mu v - f\Big)^2 \mathrm{d}\boldsymbol{s} + \int_{\partial\mathcal{D}} (v-\psi)^2 \mathrm{d}\boldsymbol{s}. \tag{2.2}$$

The main idea of PINNs is to use deep learning techniques to minimize the residual (2.2). First, it is rewritten as

$$\begin{aligned}\mathcal{J}(v) = |\mathcal{D}|\mathbb{E}_{S\sim\mathcal{U}(\mathcal{D})}\Big[&-\sum_{i,j=1}^{d}\kappa_{ij}(S)\partial_{s_i}\partial_{s_j}v(S) + \sum_{i=1}^{d}\nu_i(S)\partial_{s_i}v(S) \\ &+\mu(S)v(S) - f(S)\Big]^2 + |\partial\mathcal{D}|\mathbb{E}_{T\sim\mathcal{U}(\partial\mathcal{D})}\big[v(T)-\psi(T)\big]^2,\end{aligned}$$

where $\mathcal{U}(\mathcal{D})$ and $\mathcal{U}(\partial\mathcal{D})$ are uniform distribution on $\mathcal{D}$ and $\partial\mathcal{D}$, respectively.

To facilitate numerical computation, a discrete version of $\mathcal{J}$ is given by

$$\widehat{\mathcal{J}}(v) := \frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\Bigg[-\sum_{i,j=1}^{d}\kappa_{ij}(S_k)\partial_{s_i}\partial_{s_j}v(S_k)+\sum_{i=1}^{d}\nu_i(S_k)\partial_{s_i}v(S_k) \\ +\mu(S_k)v(S_k)-f(S_k)\Bigg]^2+\frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}\big[v(T_k)-\psi(T_k)\big]^2,$$

where $\{S_k\}_{k=1}^N$ and $\{T_k\}_{k=1}^N$ are i.i.d. Monte Carlo sample points following $\mathcal{U}(\mathcal{D})$ on $\mathcal{D}$ and $\mathcal{U}(\partial\mathcal{D})$ on $\partial\mathcal{D}$, respectively. Then, we select a deep neural network class $\mathcal{F}_{\boldsymbol{\omega}}$, within which we will minimize $\widehat{\mathcal{J}}(v_{\boldsymbol{\omega}})$ for $v_{\boldsymbol{\omega}}\in\mathcal{F}_{\boldsymbol{\omega}}$.

2.2 Neural network class

Let $D,d\in\mathbb{N}$, $H_0=d$ and $H_D=1$. Consider a neural network function $\Phi_{\boldsymbol{\omega}}:\mathbb{R}^d\to\mathbb{R}$ with following structure:

$$\begin{aligned}&\Phi^{[0]}(\boldsymbol{s})=\boldsymbol{s},\\&\Phi^{[\ell]}(\boldsymbol{s})=\sigma(\boldsymbol{W}_{\ell-1}\Phi^{[\ell-1]}(\boldsymbol{s})+\boldsymbol{c}_{\ell-1}),\quad \ell=1,\cdots,D-1,\\&\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})=\Phi^{[D]}(\boldsymbol{s})=\boldsymbol{W}_{D-1}\Phi^{[D-1]}(\boldsymbol{s})+\boldsymbol{c}_{D-1},\end{aligned}$$

where $\boldsymbol{W}_\ell=(w_{i,j}^{[\ell]})\in\mathbb{R}^{H_{\ell+1}\times H_\ell}$, $\boldsymbol{c}_\ell=(c_i^{[\ell]})\in\mathbb{R}^{H_{\ell+1}}$ and

$$\boldsymbol{\omega}=(w_{1,1}^{[0]},\cdots,w_{H_D,H_{D-1}}^{[D-1]},c_1^{[0]},\cdots,c_{H_D}^{[D-1]})\in\Omega.$$

We say $\Phi_{\boldsymbol{\omega}}$ belongs to $\mathcal{F}_\sigma(H,D,R_{\boldsymbol{\omega}})$ if it satisfies:

$$\max\{H_1,\cdots,H_D\}=H,\quad \|\boldsymbol{\omega}\|_\infty\le R_{\boldsymbol{\omega}}.$$

When solving PDEs with deep learning, the neural network must be differentiable to satisfy the PDE constraints. In this paper, we let $\sigma=\tanh$, and abbreviate the network class by $\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$.

For $\Phi_{\boldsymbol{\omega}}\in\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$, let n_ℓ be the number of its nonzero weights in the first ℓ layers. It directly follows that $n_D\le\mathfrak{G}(H,D,d)$, where

$$\mathfrak{G}(H,D,d):=(H+1)[(D-2)H+d+1]. \tag{2.3}$$

Since H typically exceeds d, we will also use the bound $n_D\le H(H+1)D$. Note that any weight vector $\boldsymbol{\omega}$ can be extended to a $\mathfrak{G}(H,D,d)$-dimensional vector $\boldsymbol{\omega}'$ by zero-padding, with $\Phi_{\boldsymbol{\omega}}=\Phi_{\boldsymbol{\omega}'}\in\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$. Therefore, the domain Ω of neural network parameters can be formalized as:

$$\Omega=[-R_{\boldsymbol{\omega}},R_{\boldsymbol{\omega}}]^{\mathfrak{G}(H,D,d)}.$$

Further, we introduce a parallel neural network class $\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}})$, defined as a linear combination of several sub-network classes $\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$. Specifically, any function $v_{q,\boldsymbol{\omega}}\in\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}})$ can be expressed as

$$v_{q,\boldsymbol{\omega}}(\boldsymbol{s})=\sum_{k=1}^{q}\delta_k\Phi_{\boldsymbol{\omega}}^k(\boldsymbol{s}),\quad \delta_k\in\mathbb{R},$$

where $\Phi_{\boldsymbol{\omega}}^k(\boldsymbol{s})\in\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$, $\sum_{k=1}^{q}|\delta_k|\leq K$. Let $\boldsymbol{\omega}_{\text{int}}^q:=(\boldsymbol{\omega}_1,\cdots,\boldsymbol{\omega}_q)$ with

$$\boldsymbol{\omega}_k=(w_{k,1,1}^{[0]},\cdots,w_{k,H_D,H_{D-1}}^{[D-1]},c_{k,1}^{[0]},\cdots,c_{k,H_D}^{[D-1]})\in\Omega$$

parameterizing $\Phi_{\boldsymbol{\omega}}^k$. Let $\boldsymbol{\omega}_{\text{ext}}^q:=(\delta_1,\cdots,\delta_q)$, $\boldsymbol{\omega}_{\text{all}}^q:=(\boldsymbol{\omega}_{\text{int}}^q,\boldsymbol{\omega}_{\text{ext}}^q)\in\Omega^q$. The symbol $\mathcal{PF}$ serves both as an abbreviation for parallel network class, and we may write $\mathcal{PF}$ instead of $\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}})$ when the context is clear.

As implied above, we will select a $\mathcal{PF}_{q,K}$ class with q sub-networks for implementation in this paper. Consequently, $\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})$ is given by

$$\begin{aligned}\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})=\frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\Big[&-\sum_{i,j=1}^{d}\kappa_{ij}(S_k)\partial_{s_i,s_j}^2v_{q,\boldsymbol{\omega}}(S_k)+\sum_{i=1}^{d}\nu_i(S_k)\partial_{s_i}v_{q,\boldsymbol{\omega}}(S_k)\\&+\mu(S_k)v_{q,\boldsymbol{\omega}}(S_k)-f(S_k)\Big]^2+\frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}\big[v_{q,\boldsymbol{\omega}}(T_k)-\psi(T_k)\big]^2.\end{aligned}\tag{2.4}$$

We now introduce some auxiliary function classes, which will play a crucial role in the subsequent analysis of optimization and statistical errors. First, let us define the squared residual classes:

$$\begin{aligned}\mathcal{E}_1':=\Big\{&\pm e:\mathcal{D}\to\mathbb{R}\,|\,\exists\, v_{q,\boldsymbol{\omega}}\in\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}}),\ s.t.\\&e(\cdot;\boldsymbol{\omega})=\Big[-\sum_{i,j=1}^{d}\kappa_{ij}\partial_{s_i,s_j}^2v_{q,\boldsymbol{\omega}}+\sum_{i=1}^{d}\nu_i\partial_{s_i}v_{q,\boldsymbol{\omega}}+\mu v_{q,\boldsymbol{\omega}}-f\Big]^2\Big\},\\\mathcal{E}_2':=\big\{&\pm e:\partial\mathcal{D}\to\mathbb{R}\,|\,\exists\, v_{q,\boldsymbol{\omega}}\in\mathcal{PF}\ s.t.\ e(\cdot;\boldsymbol{\omega})=(v_{q,\boldsymbol{\omega}}-\psi)^2\big\}.\end{aligned}$$

The corresponding residual classes without the square operation are defined as:

$$\begin{aligned}\mathcal{E}_1:=\Big\{&\pm e:\mathcal{D}\to\mathbb{R}\,|\,\exists\, v_{q,\boldsymbol{\omega}}\in\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}}),\ s.t.\\&e(\cdot;\boldsymbol{\omega})=-\sum_{i,j=1}^{d}\kappa_{ij}\partial_{s_i,s_j}^2v_{q,\boldsymbol{\omega}}+\sum_{i=1}^{d}\nu_i\partial_{s_i}v_{q,\boldsymbol{\omega}}+\mu v_{q,\boldsymbol{\omega}}-f\Big\},\\\mathcal{E}_2:=\big\{&\pm e:\partial\mathcal{D}\to\mathbb{R}\,|\,\exists\, v_{q,\boldsymbol{\omega}}\in\mathcal{PF}\ s.t.\ e(\cdot;\boldsymbol{\omega})=v_{q,\boldsymbol{\omega}}-\psi\big\}.\end{aligned}$$

Finally, we define the sub-network function classes:

$$\mathcal{E}_1^{sub} := \Big\{ \pm e : \mathcal{D} \to \mathbb{R} \,|\, \exists\, \Phi_{\boldsymbol{\omega}} \in \mathcal{F}(H, D, R_{\boldsymbol{\omega}}) \ s.t. \\ e(\cdot;\boldsymbol{\omega}) = -\sum_{i,j=1}^{d} \kappa_{ij} \partial^2_{s_i,s_j} \Phi_{\boldsymbol{\omega}} + \sum_{i=1}^{d} \nu_i \partial_{s_i} \Phi_{\boldsymbol{\omega}} + \mu \Phi_{\boldsymbol{\omega}} \Big\},$$

$$\mathcal{E}_2^{sub} := \big\{ \pm e : \partial\mathcal{D} \to \mathbb{R} \,|\, \exists\, \Phi_{\boldsymbol{\omega}} \in \mathcal{F}(H, D, R_{\boldsymbol{\omega}}) \ s.t.\ e(\cdot;\boldsymbol{\omega}) = \Phi_{\boldsymbol{\omega}} \big\}.$$

2.3 Projected stochastic gradient descent

To minimize the residual (2.4), we apply the PSGD algorithm, which is well-suited for large scale optimization problems with constraints.

We start by initializing the network parameters. The linear coefficients $\boldsymbol{\omega}_{\text{ext}}^q$ are initialized to zero, i.e., $(\boldsymbol{\omega}_{\text{ext}}^q)^{(0)} = (\delta_1, \cdots, \delta_q)^{(0)} = \mathbf{0}$; Each sub-network parameter in $(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}$ is initialized independently following a uniform distribution $\mathcal{U}[-U, U]$. Overall,

$$\big(w_{k,i,j}^{[\ell]}\big)^{(0)} \sim_{\text{i.i.d.}} \mathcal{U}[-U,U], \quad \big(c_{k,i}^{[\ell]}\big)^{(0)} \sim_{\text{i.i.d.}} \mathcal{U}[-U,U], \quad (\delta_k)^{(0)} = 0. \tag{2.5}$$

Then, for positive parameters ξ and ρ, we define the constraint sets:

(i) X_ξ: the (random) collection of weight vectors $\boldsymbol{\omega}_{\text{int}}^q$ satisfying

$$\big\|\boldsymbol{\omega}_{\text{int}}^q - (\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\big\|_2 \le \xi. \tag{2.6}$$

(ii) Y_ρ: all weight vectors $\boldsymbol{\omega}_{\text{ext}}^q$ such that

$$\big\|\boldsymbol{\omega}_{\text{ext}}^q\big\|_1 = \sum_{k=1}^{q} |\delta_k| \le \rho. \tag{2.7}$$

Since the Monte Carlo samples $\{S_k, T_k\}_{k=1}^N$ remain unchanged during optimization, $\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})$ depends only on the network parameters $\boldsymbol{\omega}_{\text{all}}^q$, and we denote it as $\mathcal{L}(\boldsymbol{\omega}_{\text{all}}^q)$:

$$\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}}) =: \mathcal{L}(\boldsymbol{\omega}_{\text{all}}^q) = \frac{1}{N}\sum_{k=1}^{N} \mathcal{L}_k(\boldsymbol{\omega}_{\text{all}}^q) = \frac{1}{N}\sum_{k=1}^{N} \mathcal{L}_k(\boldsymbol{\omega}_{\text{int}}^q, \boldsymbol{\omega}_{\text{ext}}^q).$$

Here, $\mathcal{L}_k(\boldsymbol{\omega}_{\text{all}}^q)$ represents the contribution from the k-th Monte Carlo pair $\{S_k, T_k\}$.

At each iteration $t \in [T_{\text{iter}}]$, a random index $k_t \in [N]$ is uniformly selected, and the gradient of $\mathcal{L}_{k_t}$ with respect to $\boldsymbol{\omega}_{\text{all}}^q$ is computed, followed by a gradient descent step with step size γ. The updated parameters are then projected onto $X_\xi \times Y_\rho$.

Formally, the iteration is given by

$$\begin{aligned} \boldsymbol{y}^{(t)} &= \nabla_{\boldsymbol{\omega}_{\text{all}}^q} \mathcal{L}_{k_t}\big(\boldsymbol{\omega}_{\text{int}}^{q,(t-1)}, \boldsymbol{\omega}_{\text{ext}}^{q,(t-1)}\big), \\ \big(\boldsymbol{\omega}_{\text{int}}^q, \boldsymbol{\omega}_{\text{ext}}^q\big)^{(t)} &= \text{Proj}_{X_\xi \times Y_\rho}\big\{ (\boldsymbol{\omega}_{\text{int}}^q, \boldsymbol{\omega}_{\text{ext}}^q)^{(t-1)} - \gamma \boldsymbol{y}^{(t)} \big\}. \end{aligned} \tag{2.8}$$

By [14], the projection onto the ℓ_1 ball Y_ρ can be implemented efficiently with linear time complexity relative to the dimensionality, while the projection onto the ℓ_2 ball X_ξ can be computed in closed form. It is worth emphasizing that the projection step plays a vital role in our error analysis: by constraining the neural network parameters, it enables us to bound the Rademacher complexity of the overparameterized $\mathcal{PF}$ class and derive size-independent statistical error.

Finally, we define the numerical PDE solution $v_{\mathcal{A}}$ of the PSGD algorithm as $v_{q,\boldsymbol{\omega}}$ parameterized with $(\boldsymbol{\omega}_{\text{int}}^q, \boldsymbol{\omega}_{\text{ext}}^q)^{(t_*)}$, where the optimal stopping time t_* is defined as

$$t_* := \arg\min_{t=0,\cdots,T} \mathbb{E}_{\text{SGD}}\left[\mathcal{L}(\boldsymbol{\omega}_{\text{int}}^{q,(t)}, \boldsymbol{\omega}_{\text{ext}}^{q,(t)})\right].$$

Here, $\mathbb{E}_{\text{SGD}}$ denotes the conditional expectation of the loss, taken over the random indices $\{k_t\}_{t=1}^{T_{\text{iter}}}$, given a fixed training set and network initialization $(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}$. Consequently, t_* remains a random variable depending on these conditions.

3 Complete error analysis

Building on the preliminaries above, we conduct a comprehensive error analysis that includes approximation, optimization, and statistical errors. This analysis provides guidelines for choosing the neural network architecture and optimization parameters to achieve an ϵ-accuracy between $v_{\mathcal{A}}$ and v_*. Section 3.1 introduces the error decomposition, followed by detailed analyses in Sections 3.2 to 3.4, and the main result, Theorem 3.5, in Section 3.5.

3.1 Error decomposition

We first characterize a specific 'optimization error' for the PSGD algorithm as:

$$\varepsilon_{\text{opt}} := \mathbb{E}_{\text{SGD}}\left[\widehat{\mathcal{J}}(v_{\mathcal{A}})\right] - \widehat{\mathcal{J}}(\bar{v}).$$

The term $\bar{v}$ denotes the best approximation in some parallel network class $\mathcal{PF}'$ that may not coincide with the algorithm's $\mathcal{PF}$ class, defined by:

$$\bar{v} \in \operatorname*{argmin}_{v \in \mathcal{PF}'} \|v - v_*\|^2_{\mathrm{C}^2(\bar{\mathcal{D}})}. \tag{3.1}$$

This leads to our error decomposition theorem:

Theorem 3.1. *Consider the PSGD algorithm solution $v_{\mathcal{A}}$ from Section* 2.3 *for solving Eq.* (2.1) *via PINNs, along with $\bar{v} \in \mathcal{PF}'$ specified in* (3.1)*. The $H^{1/2}$ error between $v_{\mathcal{A}}$ and the true*

solution v_* satisfies:

$$\begin{aligned}
&\mathbb{E}_{\mathrm{SGD}}\big[\|v_{\mathcal{A}}-v_*\|^2_{H^{1/2}(\mathcal{D})}\big]\\
\leq& C(\mathcal{D},d,R_0)\Big\{\underbrace{\sup_{v\in\mathcal{PF}}\big|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\big|}_{\varepsilon_{sta}}+\underbrace{\Big[\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})\Big]}_{\varepsilon_{opt}}+\underbrace{\|\bar{v}-v_*\|^2_{C^2(\bar{\mathcal{D}})}}_{\varepsilon_{app}}\Big\}.
\end{aligned}$$

Proof. By Proposition 2.2 in [26], we have

$$\begin{aligned}
&\mathbb{E}_{\mathrm{SGD}}\big[\|v_{\mathcal{A}}-v_*\|^2_{H^{1/2}(\mathcal{D})}\big]\leq C(\mathcal{D},d,R_0)\big\{\mathbb{E}_{\mathrm{SGD}}\big[\mathcal{J}(v_{\mathcal{A}})\big]-\mathcal{J}(v_*)\big\}\\
=&C(\mathcal{D},d,R_0)\Big\{\mathbb{E}_{\mathrm{SGD}}\big[\mathcal{J}(v_{\mathcal{A}})-\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]+\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})\\
&\qquad+\widehat{\mathcal{J}}(\bar{v})-\mathcal{J}(\bar{v})+\mathcal{J}(\bar{v})-\mathcal{J}(v_*)\Big\}\\
\leq&C(\mathcal{D},d,R_0)\Big\{2\sup_{v\in\mathcal{PF}}\big|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\big|+\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})+\mathcal{J}(\bar{v})-\mathcal{J}(v_*)\Big\}\\
\leq&C(\mathcal{D},d,R_0)\Big\{2\sup_{v\in\mathcal{PF}}\big|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\big|+\Big[\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})\Big]\\
&\qquad+6|\mathcal{D}|d^2R_0^2\|\bar{v}-v_*\|^2_{H^2(\mathcal{D})}+|\partial\mathcal{D}|\|\bar{v}-v_*\|^2_{C(\partial\mathcal{D})}\Big\}\\
\leq&C(\mathcal{D},d,R_0)\Big\{\underbrace{\sup_{v\in\mathcal{PF}}\big|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\big|}_{\varepsilon_{sta}}+\underbrace{\Big[\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})\Big]}_{\varepsilon_{opt}}+\underbrace{\|\bar{v}-v_*\|^2_{C^2(\bar{\mathcal{D}})}}_{\varepsilon_{app}}\Big\}.
\end{aligned}$$

This completes the proof. □

3.2 Approximation error

The approximation error ε_{app} is defined as

$$\varepsilon_{app}:=\|\bar{v}-v_*\|^2_{C^2(\bar{\mathcal{D}})},$$

where $\bar{v}$ is given in (3.1). The following approximation result in $C^2(\bar{\mathcal{D}})$ follows directly from a similar argument as in Theorem 3.2 of [28]. The proof is provided in Appendix D.

Theorem 3.2. *Given any $v_*\in C^m(\bar{\mathcal{D}})$ with $m\geq 3$, for some small $\epsilon^*>0$ and any $0<\epsilon<\epsilon^*$, there exists $v_{\bar{q},\bar{\omega}}\in\mathcal{PF}_{\bar{q},\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}})$ with*

$$\begin{aligned}
&\bar{q}=\lceil C_1\cdot\epsilon^{-\frac{d}{m-2t-2}}\rceil,\quad \bar{K}=C_2\cdot\epsilon^{-\frac{d}{m-2t-2}},\quad \bar{H}=2^{\lceil\log_2(d+m-1)\rceil+1},\\
&\bar{D}=\lceil\log_2(d+m-1)\rceil+2,\quad R_{\bar{\omega}}=C_3\cdot\epsilon^{-\frac{2d+2m}{m-2t-2}},
\end{aligned}$$

such that

$$\|v_{\bar{q},\bar{\omega}}-v_*\|_{C^2(\bar{\mathcal{D}})}\leq\epsilon,$$

where $0<\iota<1$*; Constants* C_1*,* C_2*, and* C_3 *depend exclusively on* d *and* m*.*

3.3 Optimization error

In this section, we provide a complete analysis of the optimization error ε_{opt}. Let $\bar{v}=v_{\bar{q},\bar{\omega}}\in\mathcal{PF}_{\bar{q},\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}})$ in Theorem 3.2, then ε_{opt} is defined as

$$\varepsilon_{opt}=\mathbb{E}_{\mathrm{SGD}}\left[\widehat{\mathcal{J}}(v_{\mathcal{A}})\right]-\widehat{\mathcal{J}}(v_{\bar{q},\bar{\omega}}). \tag{3.2}$$

Let $\tau>0$, and let $G,J\in\mathbb{N}$ with J sufficiently large. Set the number of sub-networks $q=\bar{q}\cdot G\cdot J$. As indicated in (3.2), the weights of $v_{\bar{q},\bar{\omega}}$ are treated as 'target parameters'. Thus, we set the sub-network width H in our implemented $v_{q,\omega}\in\mathcal{PF}$ to $\bar{H}$, the sub-network depth D to $\bar{D}$, and the uniform distribution range U to $R_{\bar{\omega}}$. For random initialization $(\boldsymbol{\omega}_{\mathrm{int}}^{q})^{(0)}=(\boldsymbol{\omega}_1^{(0)},\cdots,\boldsymbol{\omega}_q^{(0)})$, we aim to define an event which contains all the 'sufficiently good' initialization with respect to the target $(\bar{\boldsymbol{\omega}}_1,\cdots,\bar{\boldsymbol{\omega}}_{\bar{q}})$.

Let us define the event $E_{q,\bar{q},G,\tau}$ as follows: For each target parameter $\bar{\boldsymbol{\omega}}_k$, there exist at least G sub-networks among the total q sub-networks, such that their initial parameters $(\boldsymbol{\omega}_\cdot)^{(0)}$ are within a τ-neighborhood of $\bar{\boldsymbol{\omega}}_k$ in the infinity norm, i.e., $\|(\boldsymbol{\omega}_\cdot)^{(0)}-\bar{\boldsymbol{\omega}}_k\|_\infty\leq\tau$. In other words, event $E_{q,\bar{q},G,\tau}$ ensures after initialization, each target parameter $\bar{\boldsymbol{\omega}}_k$ is sufficiently approximated by a minimum of G sub-networks. We now derive the probability of $E_{q,\bar{q},G,\tau}$ with the help of its complement $E^{\mathsf{c}}_{q,\bar{q},G,\tau}$.

(i) Consider the case where $\bar{q}=G=1$ and $q=J$. The event $E^{\mathsf{c}}_{q,1,1,\tau}$ occurs when all initial weight vectors $(\boldsymbol{\omega}_i)^{(0)}$, for $i=1,\cdots,J$, deviate from $\bar{\boldsymbol{\omega}}_1$ by more than τ in the infinity norm, i.e., $\|(\boldsymbol{\omega}_i)^{(0)}-\bar{\boldsymbol{\omega}}_1\|_\infty>\tau$. Given that $\bar{\boldsymbol{\omega}}_1$ is a fixed vector in $[-R_{\bar{\omega}},R_{\bar{\omega}}]^{\mathfrak{G}(\bar{H},\bar{D},d)}$, where $\mathfrak{G}(H,D,d)$ is defined in (2.3), and the sub-network parameters are initialized independently according to $\mathcal{U}[-R_{\bar{\omega}},R_{\bar{\omega}}]$, we have

$$\mathbb{P}\left[\left\|(\boldsymbol{\omega}_i)^{(0)}-\bar{\boldsymbol{\omega}}_1\right\|_\infty\leq\tau\right]\geq\left(\frac{\tau}{2R_{\bar{\omega}}}\right)^{\mathfrak{G}(\bar{H},\bar{D},d)}\geq\left(\frac{\tau}{2R_{\bar{\omega}}}\right)^{\bar{H}(\bar{H}+1)\bar{D}}$$

for any $i\in\{1,\cdots,J\}$. Thus, we have

$$\mathbb{P}\left(E^{\mathsf{c}}_{q,1,1,\tau}\right)=\mathbb{P}\left[\forall i\in\{1,\cdots,J\}:\left\|(\boldsymbol{\omega}_i)^{(0)}-\bar{\boldsymbol{\omega}}_1\right\|_\infty>\tau\right]\leq\left[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\omega}})^{-\bar{H}(\bar{H}+1)\bar{D}}\right]^J.$$

(ii) For general $\bar{q},G\in\mathbb{N}$, $E_{q,\bar{q},G,\tau}$ occurs when each target parameter $\bar{\boldsymbol{\omega}}_k$ in the sequence $(\bar{\boldsymbol{\omega}}_1,\cdots,\bar{\boldsymbol{\omega}}_{\bar{q}})$ is approximated by at least G initial weight vectors from the set $\{(\boldsymbol{\omega}_i)^{(0)}\}_{i=1}^{\bar{q}\cdot G\cdot J}$ within a τ-neighborhood in the infinity norm. We observe that

$$E_{q,\bar{q},G,\tau}\supseteq\bigcap_{j=1}^{G}\bigcap_{k=1}^{\bar{q}}\left\{\exists i\in\left\{[(j-1)\bar{q}+k-1]J,\cdots,[(j-1)\bar{q}+k]J\right\}:\left\|(\boldsymbol{\omega}_i)^{(0)}-\bar{\boldsymbol{\omega}}_k\right\|_\infty\leq\tau\right\}.$$

Thus, it holds that

$$E^{\mathsf{c}}_{q,\bar{q},G,\tau}\subseteq\bigcup_{j=1}^{G}\bigcup_{k=1}^{\bar{q}}\Big\{\forall i\in\big\{[(j-1)\bar{q}+k-1]J,\cdots,[(j-1)\bar{q}+k]J\big\}:\big\|(\boldsymbol{\omega}_i)^{(0)}-\bar{\boldsymbol{\omega}}_k\big\|_\infty>\tau\Big\}.$$

This implies

$$\mathbb{P}\big(E^{\mathsf{c}}_{q,\bar{q},G,\tau}\big)\le\bar{q}\,G\,\mathbb{P}\big(E^{\mathsf{c}}_{q,1,1,\tau}\big)\le\bar{q}\,G\Big[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\boldsymbol{\omega}}})^{-\bar{H}(\bar{H}+1)\bar{D}}\Big]^J.$$

Therefore, we have

$$\mathbb{P}\big(E_{q,\bar{q},G,\tau}\big)=1-\mathbb{P}\big(E^{\mathsf{c}}_{q,\bar{q},G,\tau}\big)\ge1-\bar{q}\,G\Big[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\boldsymbol{\omega}}})^{-\bar{H}(\bar{H}+1)\bar{D}}\Big]^J.$$

Based on $E_{q,\bar{q},G,\tau}$, we introduce a sequence of random indices $r_{k,u}(\omega)$ which serve to identify the 'well-initialized' parameters: If $\omega\in E_{q,\bar{q},G,\tau}$, let $r_{k,u}$ be the index of the u-th sub-network among q satisfying $\|(\boldsymbol{\omega}_\cdot)^{(0)}-\bar{\boldsymbol{\omega}}_k\|_\infty\le\tau$. We also need $r_{k,u}\neq r_{k',u'}$ when $k\neq k'$ or $u\neq u'$; If $\omega\notin E_{q,\bar{q},G,\tau}$, we set $r_{k,u}=(k-1)G+u$.

A key subsequent step is to define a set of 'transition parameters' that leverage $r_{k,u}$ and $(\bar{\delta}_1,\cdots,\bar{\delta}_{\bar{q}})$ to establish a connection between $v_{\mathcal{A}}$ and the target $v_{\bar{q},\bar{\boldsymbol{\omega}}}$. Specifically, we define $\boldsymbol{\omega}^{q,*}_{\text{all}}$ as follows:

$$\boldsymbol{\omega}^{q,*}_{\text{all}}:=(\boldsymbol{\omega}^{q,*}_{\text{int}},\boldsymbol{\omega}^{q,*}_{\text{ext}}),\quad\text{where}\quad\boldsymbol{\omega}^{q,*}_{\text{int}}:=(\boldsymbol{\omega}^{q}_{\text{int}})^{(0)}.$$

For $\boldsymbol{\omega}^{q,*}_{\text{ext}}:=(\delta^*_1,\cdots,\delta^*_q)$, we set

$$\delta^*_i:=\begin{cases}\bar{\delta}_k/G & \text{if } i=r_{k,u}\text{ for }k=1,\cdots,\bar{q},\ u=1,\cdots,G,\\ 0 & \text{if } i\notin\{r_{k,u}:k=1,\cdots,\bar{q},\ u=1,\cdots,G\}.\end{cases}$$

Denoting $v_{q,\boldsymbol{\omega}}$ parameterized by $\boldsymbol{\omega}^{q,*}_{\text{all}}$ as v^*_q, we express it as:

$$v^*_q(\boldsymbol{s})=\sum_{i=1}^{q}\delta^*_i\cdot\big(\Phi^i_{\boldsymbol{\omega}}\big)^{(0)}(\boldsymbol{s})=\sum_{k,u}\delta^*_{r_{k,u}}\cdot\big(\Phi^{r_{k,u}}_{\boldsymbol{\omega}}\big)^{(0)}(\boldsymbol{s})=\sum_{k,u}\frac{\bar{\delta}_k}{G}\cdot\big(\Phi^{r_{k,u}}_{\boldsymbol{\omega}}\big)^{(0)}(\boldsymbol{s}),\tag{3.3}$$

with $(\Phi^i_{\boldsymbol{\omega}})^{(0)}$ indicating the i-th sub-network $\Phi^i_{\boldsymbol{\omega}}$ parameterized by $(\boldsymbol{\omega}_i)^{(0)}$. To clarify the connection between v^*_q and $v_{\bar{q},\bar{\boldsymbol{\omega}}}$, we expand the latter as:

$$v_{\bar{q},\bar{\boldsymbol{\omega}}}(\boldsymbol{s})=\sum_{k=1}^{\bar{q}}\bar{\delta}_k\cdot\Phi^k_{\bar{\boldsymbol{\omega}}}(\boldsymbol{s})=\sum_{k=1}^{\bar{q}}\sum_{u=1}^{G}\frac{\bar{\delta}_k}{G}\cdot\Phi^k_{\bar{\boldsymbol{\omega}}}(\boldsymbol{s}),\tag{3.4}$$

where $\Phi^k_{\bar{\boldsymbol{\omega}}}$ represents the k-th sub-network in $v_{\bar{q},\bar{\boldsymbol{\omega}}}$ with weights $\bar{\boldsymbol{\omega}}_k$. When the weights $\bar{\boldsymbol{\omega}}_k$ and $(\boldsymbol{\omega}_{r_{k,u}})^{(0)}$ are close, v^*_q approximates $v_{\bar{q},\bar{\boldsymbol{\omega}}}$ well — a property guaranteed by event $E_{q,\bar{q},G,\tau}$. The optimization error is then decomposed into two parts through $\widehat{\mathcal{J}}(v^*_q)$:

$$\varepsilon_{opt}=\underbrace{\widehat{\mathcal{J}}(v^*_q)-\widehat{\mathcal{J}}(v_{\bar{q},\bar{\boldsymbol{\omega}}})}_{\text{initialization error}}+\underbrace{\mathbb{E}_{\text{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(v^*_q)}_{\text{iteration error}}.\tag{3.5}$$

As we will see, the two error components exhibit both independence and coupling. The independence is reflected in the fact that regardless of the initial sub-network weights $(\boldsymbol{\omega}_{\text{int}}^{q})^{(0)}$ (and consequently, the initialization error), the iteration error can be rendered arbitrarily small through convex optimization techniques, provided that the iteration steps T_{iter} and over-parameterization index G are sufficiently large. Their coupling manifests in that controlling both errors at the same prescribed order with high probability necessitates specific conditions on the over-parameterization index J, which in turn constrains the selection of iteration steps T_{iter}.

Initialization error: As shown in (3.3) and (3.4), we have

$$v_q^*(\boldsymbol{s})=\sum_{k,u}\frac{\bar{\delta}_k}{G}\big(\Phi_{\boldsymbol{\omega}}^{r_{k,u}}\big)^{(0)}(\boldsymbol{s}),\quad v_{\bar{q},\bar{\boldsymbol{\omega}}}(\boldsymbol{s})=\sum_{k=1}^{\bar{q}}\bar{\delta}_k\Phi_{\bar{\boldsymbol{\omega}}}^{k}(\boldsymbol{s})=\sum_{k,u}\frac{\bar{\delta}_k}{G}\Phi_{\bar{\boldsymbol{\omega}}}^{k}(\boldsymbol{s}).$$

Denote by $e_i(\cdot;\boldsymbol{\omega})$ a function in $\mathcal{E}_i^{\text{sub}}$, $i=1,2$. Using Cauchy-Schwartz inequality and Lemma A.4, it holds that

$$\begin{aligned}
&\big|\widehat{\mathcal{J}}\big(v_q^*\big)-\widehat{\mathcal{J}}\big(v_{\bar{q},\bar{\boldsymbol{\omega}}}\big)\big|\\
\le&\frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\Big|\Big[\sum_{k,u}\frac{\bar{\delta}_k}{G}\cdot e_1\big(S_k;\boldsymbol{\omega}_{r_{k,u}}^{(0)}\big)-f(S_k)\Big]^2-\Big[\sum_{k,u}\frac{\bar{\delta}_k}{G}\cdot e_1(S_k;\bar{\boldsymbol{\omega}}_k)-f(S_k)\Big]^2\Big|\\
&+\frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}\Big|\Big[\sum_{k,u}\frac{\bar{\delta}_k}{G}\cdot e_2\big(T_k;\boldsymbol{\omega}_{r_{k,u}}^{(0)}\big)-\psi(T_k)\Big]^2-\Big[\sum_{k,u}\frac{\bar{\delta}_k}{G}\cdot e_2(T_k;\bar{\boldsymbol{\omega}}_k)-\psi(T_k)\Big]^2\Big|\\
\le&\frac{2|\mathcal{D}|}{N}\sum_{k=1}^{N}(B_1\bar{K}+R_0)\cdot\Big[\sum_{k,u}\Big|\frac{\bar{\delta}_k}{G}\Big|\cdot\big|e_1\big(S_k;\boldsymbol{\omega}_{r_{k,u}}^{(0)}\big)-e_1(S_k;\bar{\boldsymbol{\omega}}_k)\big|\Big]\\
&+\frac{2|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}(B_2\bar{K}+R_0)\cdot\Big[\sum_{k,u}\Big|\frac{\bar{\delta}_k}{G}\Big|\cdot\big|e_2\big(T_k;\boldsymbol{\omega}_{r_{k,u}}^{(0)}\big)-e_2(T_k;\bar{\boldsymbol{\omega}}_k)\big|\Big]\\
\le&2(|\mathcal{D}|+|\partial\mathcal{D}|)(L_1+L_2)(B_1\bar{K}+B_2\bar{K}+2R_0)\bar{K}\cdot\max_{k,u}\big\|\big(\boldsymbol{\omega}_{r_{k,u}}\big)^{(0)}-\bar{\boldsymbol{\omega}}_k\big\|_2.
\end{aligned}$$

Since

$$\big\|\big(\boldsymbol{\omega}_{r_{k,u}}\big)^{(0)}-\bar{\boldsymbol{\omega}}_k\big\|_2\le\big[\bar{H}(\bar{H}+1)\bar{D}\big]^{1/2}\cdot\max_{k,u}\big\|\big(\boldsymbol{\omega}_{r_{k,u}}\big)^{(0)}-\bar{\boldsymbol{\omega}}_k\big\|_\infty,$$

we get

$$\widehat{\mathcal{J}}\big(v_q^*\big)-\widehat{\mathcal{J}}\big(v_{\bar{q},\bar{\boldsymbol{\omega}}}\big)\le C_4\cdot\bar{K}^2\cdot R_{\bar{\boldsymbol{\omega}}}^{5\bar{D}}\cdot\max_{k,u}\big\|\big(\boldsymbol{\omega}_{r_{k,u}}\big)^{(0)}-\bar{\boldsymbol{\omega}}_k\big\|_\infty.$$

Therefore, with probability at least

$$1-\bar{q}G\Big[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\boldsymbol{\omega}}})^{-\bar{H}(\bar{H}+1)\bar{D}}\Big]^J,$$

the initialization error in (3.5) is bounded by

$$\widehat{\mathcal{J}}(v_q^*)-\widehat{\mathcal{J}}(v_{\bar{q},\bar{\omega}})\leq C_4\cdot\bar{K}^2\cdot R_{\bar{\omega}}^{5\bar{D}}\cdot\tau.$$

Iteration error: According to Section 2.3, we have $(\boldsymbol{\omega}_{\text{int}}^q,\boldsymbol{\omega}_{\text{ext}}^q)^{(t)}\in X_\xi\times Y_\rho$, which means $\|(\boldsymbol{\omega}_{\text{ext}}^q)^{(t)}\|_1\leq\rho$ and $\|(\boldsymbol{\omega}_{\text{int}}^q)^{(t)}-(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\|_2\leq\xi$. Also, by

$$\begin{aligned}\|(\boldsymbol{\omega}_{\text{int}}^q)^{(t)}\|_\infty&\leq\|(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\|_\infty+\|(\boldsymbol{\omega}_{\text{int}}^q)^{(t)}-(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\|_\infty\\&\leq\|(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\|_\infty+\|(\boldsymbol{\omega}_{\text{int}}^q)^{(t)}-(\boldsymbol{\omega}_{\text{int}}^q)^{(0)}\|_2,\end{aligned}$$

we have $\|(\boldsymbol{\omega}_{\text{int}}^q)^{(t)}\|_\infty\leq R_{\bar{\omega}}+\xi$. Thus,

$$v_{\mathcal{A}}\in\mathcal{PF}_{q,\rho}(\bar{H},\bar{D},R_{\bar{\omega}}+\xi).$$

The following lemma, whose proof is referred to Appendix C, is useful for bounding the iteration error.

Lemma 3.1. *Let $d_1,d_2\in\mathbb{N}$, and $U,V\geq 0$. Let $\mathrm{X}\subset\mathbb{R}^{d_1}$ and $\mathrm{Y}\subseteq\mathbb{R}^{d_2}$ be closed and convex sets. Consider the function*

$$F(\boldsymbol{x},\boldsymbol{y})=\frac{1}{N}\sum_{i=1}^N F_i(\boldsymbol{x},\boldsymbol{y}):\quad\mathbb{R}^{d_1}\times\mathbb{R}^{d_2}\to\mathbb{R}_+,$$

where for each $i\in[N]$, $F_i(\boldsymbol{x},\boldsymbol{y})$ is differentiable and $\boldsymbol{y}\mapsto F_i(\boldsymbol{x},\boldsymbol{y})$ is convex for all $\boldsymbol{x}\in\mathbb{R}^{d_1}$. Assume that for any $i\in[N]$,

$$\|\nabla_{\boldsymbol{y}}F_i(\boldsymbol{x},\boldsymbol{y})\|_2\leq V,\quad\forall(\boldsymbol{x},\boldsymbol{y})\in\mathrm{X}\times\mathrm{Y}.\tag{3.6}$$

Starting from $(\boldsymbol{x}_0,\boldsymbol{y}_0)\in\mathrm{X}\times\mathrm{Y}$, consider the iteration

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

where for each time step $t\in[T]$, the index i_t is randomly sampled from $[N]$ with uniform probability, and $\lambda=T^{-1}$. Let $\boldsymbol{y}^*\in\mathrm{Y}$ satisfy

$$|F(\boldsymbol{x}_t,\boldsymbol{y}^*)-F(\boldsymbol{x}_0,\boldsymbol{y}^*)|\leq U\|\boldsymbol{y}^*\|_2\|\boldsymbol{x}_t-\boldsymbol{x}_0\|_2\tag{3.7}$$

for all $t\in[T]$. Then, it holds that

$$\min_{t=0,\cdots T}\mathbb{E}[F(\boldsymbol{x}_t,\boldsymbol{y}_t)]-F(\boldsymbol{x}_0,\boldsymbol{y}^*)\leq U\|\boldsymbol{y}^*\|_2\,\mathrm{diam}(\mathrm{X})+\frac{\|\boldsymbol{y}^*-\boldsymbol{y}_0\|_2^2}{2}+\frac{V^2}{2T},\tag{3.8}$$

where the expectation is taken with respect to the random indices $i_1,\cdots,i_T$.

Note that $\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\|_1 \leq \bar{K}$. If we set $\rho = \bar{K}$, the 'transition parameter' $\boldsymbol{\omega}_{\text{all}}^{q,*} = (\boldsymbol{\omega}_{\text{int}}^{q,(0)}, \boldsymbol{\omega}_{\text{ext}}^{q,*})$ will fall in $X_\xi \times Y_\rho$. Letting $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ to be $(\boldsymbol{\omega}_{\text{int}}^{q}, \boldsymbol{\omega}_{\text{ext}}^{q})^{(t)}$, $\boldsymbol{y}^*$ to be $\boldsymbol{\omega}_{\text{ext}}^{q,*}$ and applying Lemma 3.1, we could directly obtain an estimate for the iteration error in (3.5). However, such application is valid only when conditions (3.6) and (3.7) are satisfied.

We first obtain an upper bound of $\nabla_{\boldsymbol{\omega}_{\text{ext}}^{q}} \mathcal{L}_k(\boldsymbol{\omega}_{\text{int}}^{q}, \boldsymbol{\omega}_{\text{ext}}^{q})$ to meet (3.6). Denote by $e_i(\cdot;\boldsymbol{\omega})$ a function in $\mathcal{E}_i^{\text{sub}}$, $i = 1,2$. Then, for any $k \in [N]$, we have

$$
\begin{aligned}
&\left\|\nabla_{\boldsymbol{\omega}_{\text{ext}}^{q}} \mathcal{L}_k\left(\boldsymbol{\omega}_{\text{int}}^{q}, \boldsymbol{\omega}_{\text{ext}}^{q}\right)\right\|_2^2 \\
=& 4\sum_{i=1}^{q}\Big\{|\mathcal{D}|\cdot e_1(S_k;\boldsymbol{\omega}_i)\cdot\Big[\textstyle\sum_{j=1}^{q}\delta_j e_1(S_k;\boldsymbol{\omega}_j) - f(S_k)\Big] \\
&\qquad + |\partial\mathcal{D}|\cdot e_2(T_k;\boldsymbol{\omega}_i)\cdot\Big[\textstyle\sum_{j=1}^{q}\delta_j e_2(T_k;\boldsymbol{\omega}_j) - \psi(T_k)\Big]\Big\}^2 \\
\leq& 8\sum_{i=1}^{q}\Big\{|\mathcal{D}|\cdot e_1(S_k;\boldsymbol{\omega}_i)\cdot\Big[\textstyle\sum_{j=1}^{q}\delta_j e_1(S_k;\boldsymbol{\omega}_j) - f(S_k)\Big]\Big\}^2 \\
&\qquad + 8\sum_{i=1}^{q}\Big\{|\partial\mathcal{D}|\cdot e_2(T_k;\boldsymbol{\omega}_i)\cdot\Big[\textstyle\sum_{j=1}^{q}\delta_j e_2(T_k;\boldsymbol{\omega}_j) - \psi(T_k)\Big]\Big\}^2 \\
\leq& 16q|\mathcal{D}|^2\cdot\sup_{e_1\in\mathcal{E}_1^{\text{sub}}}\|e_1\|_\infty^2\cdot\Big(\|\boldsymbol{\omega}_{\text{ext}}^{q}\|_1^2\cdot\sup_{e_1\in\mathcal{E}_1^{\text{sub}}}\|e_1\|_\infty^2 + R_0^2\Big) \\
&\quad + 16q|\partial\mathcal{D}|^2\cdot\sup_{e_2\in\mathcal{E}_2^{\text{sub}}}\|e_2\|_\infty^2\cdot\Big(\|\boldsymbol{\omega}_{\text{ext}}^{q}\|_1^2\cdot\sup_{e_2\in\mathcal{E}_2^{\text{sub}}}\|e_2\|_\infty^2 + R_0^2\Big) \\
\leq& 16\cdot q\cdot\bar{K}^2\left(|\mathcal{D}|^2 + |\partial\mathcal{D}|^2\right)\left[B_1^2\left(\bar{K}^2B_1^2 + R_0^2\right) + B_2^2\left(\bar{K}^2B_2^2 + R_0^2\right)\right] \\
=& C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}} + \xi)^{8\bar{D}},
\end{aligned}
$$

where we have used the properties outlined in Lemma A.4 with $R_\omega = R_{\bar{\omega}} + \xi$.

Then, we assure that $\mathcal{L}(\boldsymbol{\omega}_{\text{int}}^{q}, \boldsymbol{\omega}_{\text{ext}}^{q})$ satisfies the condition in (3.7):

$$
\begin{aligned}
&\left|\mathcal{L}\left(\boldsymbol{\omega}_{\text{int}}^{q,(t)}, \boldsymbol{\omega}_{\text{ext}}^{q,*}\right) - \mathcal{L}\left(\boldsymbol{\omega}_{\text{int}}^{q,(0)}, \boldsymbol{\omega}_{\text{ext}}^{q,*}\right)\right| \\
\leq& \frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\left|\Big[\sum_{i=1}^{q}\delta_i^* e_1(S_k;\boldsymbol{\omega}_i^{(t)}) - f(S_k)\Big]^2 - \Big[\sum_{i=1}^{q}\delta_i^* e_1(S_k;\boldsymbol{\omega}_i^{(0)}) - f(S_k)\Big]^2\right| \\
&+ \frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}\left|\Big[\sum_{i=1}^{q}\delta_i^* e_1(T_k;\boldsymbol{\omega}_i^{(t)}) - \psi(T_k)\Big]^2 - \Big[\sum_{i=1}^{q}\delta_i^* e_1(T_k;\boldsymbol{\omega}_i^{(0)}) - \psi(T_k)\Big]^2\right|.
\end{aligned}
$$

By Cauchy-Schwarz inequality and Lemma A.4, we further deduce

$$\begin{aligned}
&\left|\mathcal{L}\left(\boldsymbol{\omega}_{\text{int}}^{q,(t)},\boldsymbol{\omega}_{\text{ext}}^{q,*}\right)-\mathcal{L}\left(\boldsymbol{\omega}_{\text{int}}^{q,(0)},\boldsymbol{\omega}_{\text{ext}}^{q,*}\right)\right| \\
\leq&\frac{2|\mathcal{D}|}{N}\sum_{k=1}^{N}(B_1\bar{K}+R_0)\cdot\left\{\sum_{i=1}^{q}|\delta_i^*|^2\cdot\sum_{i=1}^{q}\left[e_1(S_k;\boldsymbol{\omega}_i^{(t)})-e_1(S_k;\boldsymbol{\omega}_i^{(0)})\right]^2\right\}^{1/2} \\
&+\frac{2|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}(B_2\bar{K}+R_0)\cdot\left\{\sum_{i=1}^{q}|\delta_i^*|^2\cdot\sum_{i=1}^{q}\left[e_2(S_k;\boldsymbol{\omega}_i^{(t)})-e_2(S_k;\boldsymbol{\omega}_i^{(0)})\right]^2\right\}^{1/2} \\
\leq&2(|\mathcal{D}|+|\partial\mathcal{D}|)(L_1+L_2)[\bar{K}(B_1+B_2)+2R_0]\cdot\left\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\right\|_2\cdot\left(\sum_{i=1}^{q}\left\|\boldsymbol{\omega}_i^{(t)}-\boldsymbol{\omega}_i^{(0)}\right\|_2^2\right)^{1/2} \\
=&C_6\cdot\bar{K}\cdot(R_{\bar{\omega}}+\xi)^{5\bar{D}}\cdot\left\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\right\|_2\cdot\left\|\boldsymbol{\omega}_{\text{int}}^{q,(t)}-\boldsymbol{\omega}_{\text{int}}^{q,(0)}\right\|_2.
\end{aligned}$$

Since

$$\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\|_2=\sqrt{\sum_{i=1}^{q}|\delta_i^*|^2}=\frac{1}{\sqrt{G}}\sqrt{\sum_{k=1}^{\bar{q}}|\bar{\delta}_k|^2}\leq\frac{1}{\sqrt{G}}\sum_{k=1}^{\bar{q}}|\bar{\delta}_k|\leq\frac{\bar{K}}{\sqrt{G}},$$

according to (3.8), the iteration error in (3.5) is bounded by

$$\begin{aligned}
&\mathbb{E}_{SGD}\left[\widehat{\mathcal{J}}(v_{\mathcal{A}})\right]-\widehat{\mathcal{J}}(v_q^*) \\
\leq&C_6\cdot\bar{K}\cdot\xi\cdot(R_{\bar{\omega}}+\xi)^{5\bar{D}}\cdot\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\|_2+\frac{1}{2}\|\boldsymbol{\omega}_{\text{ext}}^{q,*}\|_2^2+\frac{C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2T_{\text{iter}}} \\
\leq&\frac{C_6\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{5\bar{D}}\cdot\xi}{\sqrt{G}}+\frac{\bar{K}^2}{2G}+\frac{C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2T_{\text{iter}}} \\
\leq&\frac{C_7\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{5\bar{D}}\cdot\xi}{\sqrt{G}}+\frac{C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2T_{\text{iter}}}.
\end{aligned}$$

Finally, we achieve the following result.

Theorem 3.3. *Let $\tau>0$, and let $G,J\in\mathbb{N}$ with J sufficiently large. Consider the solution $v_{\mathcal{A}}\in\mathcal{PF}_{q,\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}}+\xi)$ obtained by the PSGD algorithm after T_{iter} steps, under the ℓ_1 linear coefficient constraint $\rho=\bar{K}$ and step size $\gamma=T_{\text{iter}}^{-1}$. If the number of sub-networks is set to $q=\bar{q}\cdot G\cdot J$, then with probability at least*

$$1-\bar{q}G\left[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\omega}})^{-\bar{H}(\bar{H}+1)\bar{D}}\right]^J,$$

the optimization error ε_{opt} in (3.2) is upper bounded by

$$\varepsilon_{opt}\leq C_4\cdot\bar{K}^2\cdot R_{\bar{\omega}}^{5\bar{D}}\cdot\tau+\frac{C_7\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{5\bar{D}}\cdot\xi}{\sqrt{G}}+\frac{C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2T_{\text{iter}}},$$

where ξ is the projection radius of sub-network weights, while C_4, C_5, and C_7 are universal constants depending only on $\mathcal{D}$, $\bar{H}$, $\bar{D}$, d, and R_0.

3.4 Statistical error

We define the statistical error as

$$\varepsilon_{sta} := \sup_{v\in\mathcal{PF}} \left|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\right|.$$

Our goal is to bound ε_{sta} with high probability, for which we first analyze its expectation $\mathbb{E}[\varepsilon_{sta}]$. The following lemma is direct.

Lemma 3.2. *The expected statistical error decomposes as:*

$$\begin{aligned}\mathbb{E}[\varepsilon_{sta}] &= \mathbb{E}_{\{S_k,T_k\}_{k=1}^N}\left[\sup_{v_{q,\boldsymbol{\omega}}\in\mathcal{PF}}\left|\mathcal{J}(v_{q,\boldsymbol{\omega}})-\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})\right|\right]\\ &\le \sum_{i=1}^{2}\mathbb{E}_{\{S_k,T_k\}_{k=1}^N}\sup_{v_{q,\boldsymbol{\omega}}\in\mathcal{PF}}\left|\mathcal{J}_i(v_{q,\boldsymbol{\omega}})-\widehat{\mathcal{J}}_i(v_{q,\boldsymbol{\omega}})\right| = \sum_{i=1}^{2}\mathbb{E}[\varepsilon_{sta}^i],\end{aligned}$$

where $\mathcal{J}_1$ quantifies the PDE residual over $\mathcal{D}$:

$$\begin{aligned}\mathcal{J}_1(v_{q,\boldsymbol{\omega}}) = |\mathcal{D}|\mathbb{E}_{S\sim\mathcal{U}(\mathcal{D})}\Bigg[&-\sum_{i,j=1}^{d}\kappa_{ij}(S)\partial_{s_i}\partial_{s_j}v_{q,\boldsymbol{\omega}}(S)+\sum_{i=1}^{d}\nu_i(S)\partial_{s_i}v_{q,\boldsymbol{\omega}}(S)\\ &+\mu(S)v_{q,\boldsymbol{\omega}}(S)-f(S)\Bigg]^2,\end{aligned}$$

$\mathcal{J}_2$ measures the boundary condition mismatch:

$$\mathcal{J}_2(v_{q,\boldsymbol{\omega}}) = |\partial\mathcal{D}|\mathbb{E}_{T\sim\mathcal{U}(\partial\mathcal{D})}\left[v_{q,\boldsymbol{\omega}}(T)-\psi(T)\right]^2,$$

and their discrete counterparts $\widehat{\mathcal{J}}_i$ are denote by:

$$\begin{aligned}\widehat{\mathcal{J}}_1(v_{q,\boldsymbol{\omega}}) = \frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\Bigg[&-\sum_{i,j=1}^{d}\kappa_{ij}(S_k)\partial_{s_i}\partial_{s_j}v_{q,\boldsymbol{\omega}}(S_k)+\sum_{i=1}^{d}\nu_i(S_k)\partial_{s_i}v_{q,\boldsymbol{\omega}}(S_k)\\ &+\mu(S_k)v_{q,\boldsymbol{\omega}}(S_k)-f(S_k)\Bigg]^2,\\ \widehat{\mathcal{J}}_2(v_{q,\boldsymbol{\omega}}) = \frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}&\left[v_{q,\boldsymbol{\omega}}(T_k)-\psi(T_k)\right]^2.\end{aligned}$$

To control each $\mathbb{E}[\varepsilon_{sta}^i]$, we use Rademacher complexity as our main analytical tool.

Definition 3.1. For a function class $\mathcal{F}$ and a random sample $\{Z_k\}_{k=1}^N$, we characterize two variants of Rademacher complexity:

$$\mathcal{R}_N(\mathcal{F}) = \mathbb{E}_{\{Z_k,\eta_k\}_{k=1}^N}\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{k=1}^{N}\eta_k f(Z_k)\right],\quad \bar{\mathcal{R}}_N(\mathcal{F}) = \mathbb{E}_{\{Z_k,\eta_k\}_{k=1}^N}\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\left|\sum_{k=1}^{N}\eta_k f(Z_k)\right|\right],$$

where $\{\eta_k\}_{k=1}^N$ are i.i.d. Rademacher random variables.

Our analysis relies on standard tools from statistical learning theory, such as Talagrand's contraction inequality (see, e.g., [40]). The following lemma connects $\mathbb{E}[\varepsilon_{sta}^i]$ to the Rademacher complexity of specific function classes derived from our model $\mathcal{PF}$.

Lemma 3.3. *Let $\mathcal{PF}=\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}})$. Recall the auxiliary function classes defined in Section 2.2. Then, it holds that*

$$\mathbb{E}[\varepsilon_{sta}^1]\leq 16|\mathcal{D}|d^2R_0K^2DH^{2D-2}R_{\boldsymbol{\omega}}^{2D}\cdot\bar{\mathcal{R}}_N(\mathcal{E}_1^{\text{sub}}),$$
$$\mathbb{E}[\varepsilon_{sta}^2]\leq 2|\partial\mathcal{D}|\big[K(H+1)R_{\boldsymbol{\omega}}+2R_0\big]K\cdot\bar{\mathcal{R}}_N(\mathcal{E}_2^{\text{sub}}).$$

Proof. Note that $\mathcal{E}_1'$ and $\mathcal{E}_2'$ are exactly the function classes corresponding to the squared loss terms in $\mathcal{J}_1$ and $\mathcal{J}_2$. Using standard symmetrization arguments (see, e.g., [48], Chapter 4), we conclude that

$$\mathbb{E}[\varepsilon_{sta}^1]\leq 2|\mathcal{D}|\cdot\mathcal{R}_N(\mathcal{E}_1')\quad\text{and}\quad\mathbb{E}[\varepsilon_{sta}^2]\leq 2|\partial\mathcal{D}|\cdot\mathcal{R}_N(\mathcal{E}_2').$$

The core of the proof is to relate $\mathcal{R}_N(\mathcal{E}_i')$ to the complexity of the much simpler sub-network class $\mathcal{E}_i^{\text{sub}}$. We show the derivation for the first term.

Step 1: From squared loss to linear loss. Let $e(\cdot;\boldsymbol{\omega})\in\mathcal{E}_1$ and $\tilde{e}(\cdot;\boldsymbol{\omega})=e(\cdot;\boldsymbol{\omega})^2\in\mathcal{E}_1'$. The function $\Phi(x)=x^2$ is Lipschitz on the range of e. Specifically, for any two functions $e_1,e_2\in\mathcal{E}_1$, we have $|e_1^2-e_2^2|=|e_1+e_2||e_1-e_2|$. The term $|e_1+e_2|$ can be bounded using the properties of the function class $\mathcal{PF}$ and Lemma A.4. This leads to a Lipschitz constant for Φ, which by Talagrand's contraction inequality (see [40], Lemma 5.7) yields:

$$\mathcal{R}_N(\mathcal{E}_1')\leq 8d^2R_0DKR_{\boldsymbol{\omega}}^{2D}H^{2D-2}\cdot\mathcal{R}_N(\mathcal{E}_1).$$

A similar bound holds for the boundary term:

$$\mathcal{R}_N(\mathcal{E}_2')\leq 2[K(H+1)R_{\boldsymbol{\omega}}+R_0]\cdot\mathcal{R}_N(\mathcal{E}_2).$$

Step 2: From the full network class to the sub-network class. Finally, we relate $\mathcal{R}_N(\mathcal{E}_i)$ to $\bar{\mathcal{R}}_N(\mathcal{E}_i^{\text{sub}})$. A function $e\in\mathcal{E}_1$ has the form $e(\boldsymbol{s})=\sum_{j=1}^q\delta_j e_*(\boldsymbol{s};\boldsymbol{\omega}_j)-f(\boldsymbol{s})$, where $e_*\in\mathcal{E}_1^{\text{sub}}$ and $\sum_j|\delta_j|\leq K$. By the definition of Rademacher complexity and properties of the supremum, we get:

$$\begin{aligned}\mathcal{R}_N(\mathcal{E}_1)&=\mathbb{E}_{\{S_k,\eta_k\}}\left[\sup_{\boldsymbol{\omega}_{\text{all}}^q}\frac{1}{N}\sum_{k=1}^N\eta_k\left(\sum_{j=1}^q\delta_j e_*(S_k;\boldsymbol{\omega}_j)-f(S_k)\right)\right]\\&\leq\mathbb{E}_{\{S_k,\eta_k\}}\left[\sup_{\boldsymbol{\omega}_{\text{all}}^q}\sum_{j=1}^q|\delta_j|\cdot\left|\frac{1}{N}\sum_{k=1}^N\eta_k e_*(S_k;\boldsymbol{\omega}_j)\right|\right]\\&\leq K\cdot\mathbb{E}_{\{S_k,\eta_k\}}\left[\sup_{e_*\in\mathcal{E}_1^{\text{sub}}}\left|\frac{1}{N}\sum_{k=1}^N\eta_k e_*(S_k)\right|\right]=K\cdot\bar{\mathcal{R}}_N(\mathcal{E}_1^{\text{sub}}).\end{aligned}$$

Similarly, $\mathcal{R}_N(\mathcal{E}_2)\leq K\cdot\bar{\mathcal{R}}_N(\mathcal{E}_2^{\text{sub}})$. Combining the results from both steps yields the statement of the lemma. □

Remark 3.1. This conclusion reveals an important fact: the Rademacher complexity of the full loss function class is controlled by K and the complexity of the elementary sub-network class $\mathcal{E}_i^{\text{sub}}$. This suggests that the network's overall complexity may not be affected by the number of sub-networks q, aiding us in managing the statistical error within the over-parameterized setting, where q can grow arbitrarily large.

Then, we introduce the definition of 'covering number' and Proposition 3.1, known as Dudley's entropy integral theorem (see [15]), which provides an effective tool to control $\bar{\mathcal{R}}_N(\mathcal{E}_i^{\text{sub}})$, $i=1,2$.

Definition 3.2. 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 a $t_c \in T_c$ 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.

Proposition 3.1. For any function class $\mathcal{F}$ mapping from $\mathcal{D}$ to $\mathbb{R}$ that contains the zero function and $\|u\|_{L^\infty(\mathcal{D})} \leq \mathcal{B}$ for all $u \in \mathcal{F}$, it holds that

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

Based on above results, we present the following lemma, providing an upper bound for each $\bar{\mathcal{R}}_N(\mathcal{E}_i^{\text{sub}})$,$i=1,2$ in terms of the covering number of $\mathcal{E}_i^{\text{sub}}$.

Lemma 3.4. *For $i=1,2$, we have*

$$\bar{\mathcal{R}}_N(\mathcal{E}_i^{\text{sub}}) \leq C(H,D,d,R_0) R_{\boldsymbol{\omega}}^{2D} \cdot N^{-1/2} \cdot \sqrt{\log(R_{\boldsymbol{\omega}} HDN)}.$$

Proof. Using Proposition 3.1, for $i=1,2$,

$$\bar{\mathcal{R}}_N(\mathcal{E}_i^{\text{sub}}) \leq \inf_{0<\delta<B_i} \left[4\delta + \frac{12}{\sqrt{N}} \int_\delta^{B_i} \sqrt{\log \mathcal{C}(\epsilon,\mathcal{E}_i^{\text{sub}},\|\cdot\|_{L^\infty})} \, \mathrm{d}\epsilon \right].$$

Applying Lemma 5.6 in [27], we have

$$\mathcal{C}(\epsilon,\mathcal{E}_i^{\text{sub}},\|\cdot\|_{L^\infty}) \leq \mathcal{C}(\epsilon L_i^{-1},\Omega,\|\cdot\|_2).$$

By Lemma 5.5 in [27], it further holds that

$$\mathcal{C}(\epsilon L_i^{-1},\Omega,\|\cdot\|_2) \leq \left[\frac{2R_{\boldsymbol{\omega}} \sqrt{H(H+1)D} \cdot L_i}{\epsilon} \right]^{H(H+1)D}.$$

The above B_i and L_i denote the boundedness and smoothness indices of $\mathcal{E}_i^{\text{sub}}$ with respect to model parameters (see Lemma A.4 for details). For instance, to upper bound

$\bar{\mathcal{R}}_N(\mathcal{E}_1^{\rm sub})$, we substitute B_1 and L_1 from Lemma A.4 and obtain

$$\begin{aligned}\bar{\mathcal{R}}_N(\mathcal{E}_1^{\rm sub}) &\leq \inf_{0<\delta<B_1}\left[4\delta+\frac{12}{\sqrt{N}}\int_\delta^{B_1}\sqrt{\log\mathcal{C}(\epsilon,\mathcal{E}_1^{\rm sub},\|\cdot\|_{L^\infty})}\mathrm{d}\epsilon\right]\\ &\leq \inf_{0<\delta<B_1}\left[4\delta+12H\sqrt{D}\cdot B_1 N^{-1/2}\log^{1/2}\left(44d^2D^3R_0H^{3D+2}R_{\boldsymbol{\omega}}^{2D}\delta^{-1}\right)\right].\end{aligned}$$

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

$$\bar{\mathcal{R}}_N(\mathcal{E}_1^{\rm sub})\leq 36R_0H^{2D}d^2D^2(3D+2)\cdot R_{\boldsymbol{\omega}}^{2D}\cdot N^{-1/2}\cdot\sqrt{\log(R_{\boldsymbol{\omega}}HDN)}.$$

Likewise, $\bar{\mathcal{R}}_N(\mathcal{E}_2^{\rm sub})$ is bounded similarly to $\bar{\mathcal{R}}_N(\mathcal{E}_1^{\rm sub})$. □

Leveraging all preceding analyses, we can ultimately establish probabilistic bounds on ε_{sta} with high confidence:

Theorem 3.4. *Let $\mathcal{PF}=\mathcal{PF}_{q,K}(H,D,R_{\boldsymbol{\omega}})$. For $0<\zeta<1$, with probability at least $1-\zeta$, it holds that*

$$\begin{aligned}\varepsilon_{sta}&=\sup_{v_{q,\boldsymbol{\omega}}\in\mathcal{PF}}\left|\mathcal{J}(v_{q,\boldsymbol{\omega}})-\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})\right|\\ &\leq C_8\cdot K^2R_{\boldsymbol{\omega}}^{4D}N^{-1/2}\left(\sqrt{\log(R_{\boldsymbol{\omega}}HDN)}+\sqrt{\log\zeta^{-1}}\right),\end{aligned}$$

where C_8 is a universal constant which only depends on $\mathcal{D},H,D,d$ and R_0.

Proof. By Lemma 3.2 and Lemma 3.3, we have that

$$\mathbb{E}[\varepsilon_{sta}]\leq 16|\mathcal{D}|d^2R_0K^2DR_{\boldsymbol{\omega}}^{2D}H^{2D-2}\cdot\bar{\mathcal{R}}_N(\mathcal{E}_1^{\rm sub})+2|\partial\mathcal{D}|[K(H+1)R_{\boldsymbol{\omega}}+2R_0]K\cdot\bar{\mathcal{R}}_N(\mathcal{E}_2^{\rm sub}).$$

By Lemma 3.4, for $i=1,2$,

$$\bar{\mathcal{R}}_N(\mathcal{E}_i^{\rm sub})\leq C(H,D,d,R_0)R_{\boldsymbol{\omega}}^{2D}\cdot N^{-1/2}\cdot\sqrt{\log(R_{\boldsymbol{\omega}}HDN)},$$

then we can get

$$\mathbb{E}[\varepsilon_{sta}]\leq C_9(\mathcal{D},R_0,d,H,D)\cdot K^2R_{\boldsymbol{\omega}}^{4D}N^{-1/2}\sqrt{\log(R_{\boldsymbol{\omega}}HDN)}. \tag{3.9}$$

Now we define

$$\varphi(S_1,\cdots,S_N,T_1,\cdots,T_N):=\sup_{v_{q,\boldsymbol{\omega}}\in\mathcal{PF}}|\mathcal{J}(v_{q,\boldsymbol{\omega}})-\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})|=\varepsilon_{sta}.$$

Further, we expand $\mathcal{J}(v_{q,\boldsymbol{\omega}})$ as follows

$$\mathcal{J}(v_{q,\boldsymbol{\omega}})=|\mathcal{D}|\mathbb{E}_{S\sim\mathcal{U}(\mathcal{D})}\left[\tilde{e}(S;\boldsymbol{\omega}_{\rm all}^q)\right]+|\partial\mathcal{D}|\mathbb{E}_{T\sim\mathcal{U}(\partial\mathcal{D})}\left[\hat{e}(T;\boldsymbol{\omega}_{\rm all}^q)\right],$$

where $\tilde{e}(\cdot;\boldsymbol{\omega})$ and $\hat{e}(\cdot;\boldsymbol{\omega})$ denote a function in $\mathcal{E}_1'$ and $\mathcal{E}_2'$, respectively. Also, we expand $\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})$ as follows

$$\widehat{\mathcal{J}}(v_{q,\boldsymbol{\omega}})=\frac{|\mathcal{D}|}{N}\sum_{k=1}^{N}\tilde{e}(S_k;\boldsymbol{\omega}_{\text{all}}^q)+\frac{|\partial\mathcal{D}|}{N}\sum_{k=1}^{N}\hat{e}(T_k;\boldsymbol{\omega}_{\text{all}}^q).$$

We then examine the difference of $\varphi(S_1,\cdots,S_N,T_1,\cdots,T_N)$:

$$\begin{aligned}
&|\varphi(S_1,\cdots,S_i,\cdots,T_N)-\varphi(S_1,\cdots,S_i',\cdots,T_N)|\\
\leq&\frac{|\mathcal{D}|}{N}\sup_{\boldsymbol{\omega}_{\text{all}}^q\in\Omega^q}\left|\tilde{e}(S_i;\boldsymbol{\omega}_{\text{all}}^q)-\tilde{e}(S_i';\boldsymbol{\omega}_{\text{all}}^q)\right|\\
=&\frac{|\mathcal{D}|}{N}\sup_{\boldsymbol{\omega}_{\text{all}}^q\in\Omega^q}\left|\Big[\sum_{j=1}^{q}\delta_j e_*(S_i;\boldsymbol{\omega}_j)-f(S_i)\Big]^2-\Big[\sum_{j=1}^{q}\delta_j e_*(S_i';\boldsymbol{\omega}_j)-f(S_i')\Big]^2\right|\\
\leq&32|\mathcal{D}|N^{-1}d^4R_0^2K^2D^2H^{4D-4}R_{\boldsymbol{\omega}}^{4D},
\end{aligned}$$

where $e_*(\cdot;\boldsymbol{\omega})$ denotes a function in $\mathcal{E}_1^{\text{sub}}$, and we have used the boundedness properties outlined in Lemma A.4. Similarly, we have

$$|\varphi(S_1,\cdots,T_j,\cdots,T_N)-\varphi(S_1,\cdots,T_j',\cdots,T_N)|\leq 8|\partial\mathcal{D}|N^{-1}R_0^2K^2(H+1)^2R_{\boldsymbol{\omega}}^2.$$

Then, by McDiarmid's inequality (see [37]), we have

$$\begin{aligned}
\varepsilon_{sta}\leq&\mathbb{E}[\varepsilon_{sta}]+\epsilon\\
\leq&C_9(\mathcal{D},R_0,d,H,D)\cdot K^2R_{\boldsymbol{\omega}}^{4D}N^{-1/2}\sqrt{\log(R_{\boldsymbol{\omega}}HDN)}+\epsilon
\end{aligned}$$

with probability as least

$$1-2\exp\left\{-\frac{N\epsilon^2}{C_{10}d^4(|\partial\mathcal{D}|^4+|\mathcal{D}|^2)R_0^4H^{8D-8}R_{\boldsymbol{\omega}}^{8D}K^4R_0^4D^4}\right\}.$$

This implies that with probability at least $1-\zeta$, it holds that

$$\varepsilon_{sta}\leq C_8(\mathcal{D},R_0,d,H,D)\cdot K^2R_{\boldsymbol{\omega}}^{4D}N^{-1/2}\Big(\sqrt{\log(R_{\boldsymbol{\omega}}HDN)}+\sqrt{\log\zeta^{-1}}\Big).$$

This completes the proof. □

Remark 3.2. Theorem 3.4 provides a statistical error analysis for general $\mathcal{PF}$ classes with a notable property: the error bound remains independent of the sub-network count q. This feature proves valuable in over-parameterized settings where q may grow arbitrarily large. Since Theorem 3.3 establishes that $v_{\mathcal{A}}\in\mathcal{PF}_{q,\bar{K}}(\bar{H},\bar{D},R_{\bar{\boldsymbol{\omega}}}+\xi)$, combining these results yields more refined bounds on $\mathcal{E}_{sta}$.

3.5 Main result

After analyzing the approximation, optimization, and statistical errors individually, we present our main theorem, which combines these components to establish the total error bound for PINNs in solving (2.1). This result addresses the question of how to determine the appropriate number of training samples, key architectural parameters of the neural networks, step size for the projected stochastic gradient descent optimization, and the required number of iterations to ensure that the gradient descent process closely approximates the true solution of (2.1) to a specified precision.

Theorem 3.5. *To solve the elliptic PDE* (2.1) *using PINNs, we employ the $\mathcal{PF}$ architecture from Section* 2.2 *which consists of q sub-networks, each with width H and depth D. Sub-network parameters are initialized uniformly on $[-B,B]$ via* (2.5)*. We optimize using the PSGD algorithm from Section* 2.3*, with projection radii ξ and ρ defined in* (2.6) *and* (2.7)*. Let the Monte Carlo sample size in* (2.4) *be N. Let $v_{\mathcal{A}}$ be the solution of the PSGD algorithm with iteration steps T_{iter} and step size γ. For any $0<\epsilon\ll 1$, set*

$$
\begin{aligned}
&q=\lceil C\cdot\epsilon^{-\tilde{C}_1(\iota,d,\varsigma_0,m)}\rceil, \quad && H=2^{\lceil\log_2(d+m-1)\rceil+1}, \quad && D=\lceil\log_2(d+m-1)\rceil+2,\\
&B=C\cdot\epsilon^{-\frac{2d+2m}{m-2\iota-2}}, && \xi=\epsilon^{-\varsigma}, && \rho=C\cdot\epsilon^{-\frac{d}{m-2\iota-2}},\\
&T_{\text{iter}}=C\cdot\epsilon^{-\tilde{C}_2(\iota,d,\varsigma_0,m)}, && \gamma=C\cdot\epsilon^{\tilde{C}_2(\iota,d,\varsigma_0,m)}, && N=\lceil C\cdot\epsilon^{-\tilde{C}_3(\iota,d,\varsigma_0,m)}\rceil.
\end{aligned}
$$

Suppose that $v_*\in C^m(\bar{\mathcal{D}})$ for $m\geq 3$ is the target solution of Eq. (2.1)*. Then, with probability at least $1-2\cdot\epsilon^{\tilde{C}_3(\iota,d,\varsigma_0,m)}$, the total error*

$$
\mathbb{E}_{SGD}\left[\|v_{\mathcal{A}}-v_*\|^2_{H^{1/2}(\mathcal{D})}\right]\leq C\epsilon\log^{1/2}(C\epsilon^{-1})=\tilde{\mathcal{O}}(\epsilon),
$$

where

$$
\begin{aligned}
&\varsigma>0, \quad \varsigma_0=\max\{\varsigma,(2d+2m)(m-2\iota-2)^{-1}\},\\
&\tilde{C}_1(\iota,d,\varsigma_0,m)=\frac{C'\cdot(d+m)^3\log^2(d+m-1)}{m-2\iota-2}+19\varsigma_0\log_2(d+m-1)+60\varsigma_0,\\
&\tilde{C}_2(\iota,d,\varsigma_0,m)=\frac{C''\cdot(d+m)^3\log^2(d+m-1)}{m-2\iota-2}+27\varsigma_0\log_2(d+m-1)+84\varsigma_0,\\
&\tilde{C}_3(\iota,d,\varsigma_0,m)=8\varsigma_0\log_2(d+m-1)+\frac{4d}{m-2\iota-2}+24\varsigma_0+2.
\end{aligned}
$$

Meanwhile, C denotes a universal constant which is defined place by place and only depends on $\mathcal{D},H,D,d,R_0$ and m; C' and C'' are positive constants; $0<\iota<1$ is an arbitrarily small positive number.

Proof. Reviewing the error decomposition in Theorem 3.1, we have

$$\begin{aligned}&\mathbb{E}_{\mathrm{SGD}}\big[\|v_{\mathcal{A}}-v_*\|^2_{H^{1/2}(\mathcal{D})}\big]\\ \leq& C(\mathcal{D},d,R_0)\Big\{\underbrace{\sup_{v\in\mathcal{PF}}\big|\mathcal{J}(v)-\widehat{\mathcal{J}}(v)\big|}_{\varepsilon_{sta}}+\underbrace{\mathbb{E}_{\mathrm{SGD}}\big[\widehat{\mathcal{J}}(v_{\mathcal{A}})\big]-\widehat{\mathcal{J}}(\bar{v})}_{\varepsilon_{opt}}+\underbrace{\|\bar{v}-v_*\|^2_{C^2(\bar{\mathcal{D}})}}_{\varepsilon_{app}}\Big\}.\end{aligned}$$

We can divide the proof of the theorem into the following three steps:

Step 1: According to Theorem 3.2, we know that for any $\epsilon>0$, there exists a neural network $v_{\bar{q},\bar{\omega}}\in\mathcal{PF}_{\bar{q},\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}})$, with

$$\begin{aligned}&\bar{q}=\lceil C_1\cdot\epsilon^{-\frac{d}{m-2\iota-2}}\rceil,\quad \bar{K}=C_2\cdot\epsilon^{-\frac{d}{m-2\iota-2}},\quad \bar{H}=2^{\lceil\log_2(d+m-1)\rceil+1},\\ &\bar{D}=\lceil\log_2(d+m-1)\rceil+2,\quad R_{\bar{\omega}}=C_3\cdot\epsilon^{-\frac{2d+2m}{m-2\iota-2}},\end{aligned}$$

such that the approximation error $\varepsilon_{app}\leq C\epsilon^2\leq C\epsilon$.

Step 2: By Theorem 3.4, with probability at least $1-\zeta$, the statistical error satisfies

$$\varepsilon_{sta}\leq C_8\cdot\bar{K}^2(R_{\bar{\omega}}+\xi)^{4\bar{D}}N^{-1/2}\cdot\big\{\log^{1/2}[(R_{\bar{\omega}}+\xi)\bar{H}\bar{D}N]+\log^{1/2}(\zeta^{-1})\big\}.$$

Setting

$$N=\lceil C\cdot\epsilon^{-\tilde{C}_3(\iota,d,\varsigma_0,m)}\rceil,\quad \zeta=C\cdot\epsilon^{\tilde{C}_3(\iota,d,\varsigma_0,m)}$$

with

$$\begin{aligned}&\varsigma_0=\max\{\varsigma,(2d+2m)(m-2\iota-2)^{-1}\},\\ &\tilde{C}_3=8\varsigma_0\log_2(d+m-1)+\frac{4d}{m-2\iota-2}+24\varsigma_0+2,\end{aligned}$$

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

Step 3: In order to bound the optimization error $\varepsilon_{opt}\leq C\epsilon$ with probability at least $1-\zeta$, we need to determine parameters $G,J,\tau,T_{\mathrm{iter}},\rho$ such that the following inequalities hold

$$1-\bar{q}G\Big[1-\tau^{\bar{H}(\bar{H}+1)\bar{D}}(2R_{\bar{\omega}})^{-\bar{H}(\bar{H}+1)\bar{D}}\Big]^J\geq 1-\zeta,\tag{3.10}$$

$$C_4\bar{K}^2R_{\bar{\omega}}^{5\bar{D}}\tau+\frac{C_7\bar{K}^2(R_{\bar{\omega}}+\xi)^{5\bar{D}}\xi}{\sqrt{G}}+\frac{C_5q\bar{K}^2(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2T_{\mathrm{iter}}}\leq C\epsilon.\tag{3.11}$$

To ensure the first and third terms in (3.11) are bounded by $C\epsilon$, we require

$$\begin{aligned}&\tau\leq C\cdot\epsilon^{\frac{10(d+m)\log_2(d+m-1)+23d+20m}{m-2\iota-2}+1},\\ &G\geq\lceil C\cdot\epsilon^{-10\log_2(d+m-1)\varsigma_0-32\varsigma_0-\frac{6d}{m-2\iota-2}-2}\rceil.\end{aligned}$$

Since $1-x\leq\exp(-x)$, and $\exp(-x)\leq x^{-2}$ for $x\geq 0$, we could solve for J as follows:

$$J\geq\sqrt{\frac{\bar{q}G}{\zeta}}\left(\frac{\tau}{2R_{\bar{\omega}}}\right)^{-\bar{H}(\bar{H}+1)\bar{D}}\Rightarrow J\geq\lceil C\epsilon^{-\frac{C_0(d+m)^3\log^2(d+m-1)}{m-2\iota-2}-9\varsigma_0\log_2(d+m-1)-28\varsigma_0}\rceil,$$

where C_0 is a positive constant. Then, $q=\bar{q}\cdot G\cdot J\geq\lceil C\cdot\epsilon^{-\tilde{C}_1(\iota,d,\varsigma_0,m)}\rceil$. Finally, we turn to the second term of (3.11). To ensure that

$$\frac{C_5\cdot q\cdot\bar{K}^2\cdot(R_{\bar{\omega}}+\xi)^{8\bar{D}}}{2\,T_{\text{iter}}}\leq C\epsilon,$$

we require

$$T_{\text{iter}}\geq\lceil C\cdot\epsilon^{-\tilde{C}_2(\iota,d,\varsigma_0,m)}\rceil.$$

By Theorem 3.3, $\gamma=T_{\text{iter}}^{-1}$. Thus, we complete the proof. □

Remark 3.3. The projection radii in our method, ξ and ρ, diverge with the target accuracy ϵ, permitting significant parameter updates during optimization.This behavior is distinct from both the minimal parameter changes inherent to NTK theory [24] and the static, frozen inner-layer parameters of random feature models [10].

Remark 3.4. Our convergence analysis relies on a sufficiently large number of subnetworks, q (over-parameterization), to guarantee that with high probability, a randomly initialized subnetwork is close enough to the target function for optimization to succeed. This leads to a theoretical requirement where q scales polynomially with the inverse accuracy, ϵ^{-1}. It is crucial to recognize this as a 'worst-case' sufficiency condition that may be looser than what is needed in practice. In line with the broader deep learning literature, practitioners should view this bound as a foundational guideline for model scaling, not a strict prescription. A dedicated empirical study of this scaling relationship remains a valuable direction for our future research.

4 Conclusion

In this paper, we provide the first complete error analysis for PINNs that includes the approximation error, statistical error, and optimization error in the scenario of over-parameterization. Our analysis is based on the PSGD algorithm and does not require constraining the neural network weights near their initial values during the optimization process. This marks a milestone in the field of theoretical understanding of solving PDEs via PINNs.

Acknowledgments

This work is supported 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.

A Some properties of the neural networks

Lemmas A.1–A.4 establish several properties of $\mathcal{F}(H,D,R_\omega)$ and $\mathcal{E}_i^{\text{sub}}$ defined in Section 2.2. Lemmas A.1 and A.2 can be found in [27], and the proof of Lemma A.3 is provided in Appendix B. Below, we always assume that $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$ are two arbitrary parameter vectors satisfying

$$\boldsymbol{\omega},\tilde{\boldsymbol{\omega}}\in\Omega=[-R_\omega,R_\omega]^{\mathfrak{G}(H,D,d)},\quad R_\omega\geq 1.$$

Lemma A.1. *For any $\Phi_{\boldsymbol{\omega}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$, we have $|\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})|\leq(H+1)R_\omega$. Moreover, for any $\Phi_{\boldsymbol{\omega}},\Phi_{\tilde{\boldsymbol{\omega}}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$,*

$$|\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})-\Phi_{\tilde{\boldsymbol{\omega}}}(\boldsymbol{s})|\leq 2H^D\sqrt{D}\cdot R_\omega^{D-1}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2.$$

Lemma A.2. *For any $\Phi_{\boldsymbol{\omega}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$, we have*

$$\left|\partial_{s_\alpha}\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})\right|\leq H^{D-1}R_\omega^D.$$

Moreover, for any $\Phi_{\boldsymbol{\omega}},\Phi_{\tilde{\boldsymbol{\omega}}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$, $\alpha\in[d]$,

$$|\partial_{s_\alpha}\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})-\partial_{s_\alpha}\Phi_{\tilde{\boldsymbol{\omega}}}(\boldsymbol{s})|\leq 4H^{2D-1}D^{3/2}\cdot R_\omega^{2D}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2.$$

Lemma A.3. *For any $\Phi_{\boldsymbol{\omega}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$, we have*

$$\left|\partial_{s_\alpha}\partial_{s_\beta}\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})\right|\leq DH^{2D-2}R_\omega^{2D}.$$

Moreover, for any $\Phi_{\boldsymbol{\omega}},\Phi_{\tilde{\boldsymbol{\omega}}}\in\mathcal{F}(H,D,R_\omega)$ and $\boldsymbol{s}\in\mathcal{D}$, $\alpha,\beta\in[d]$,

$$|\partial_{s_\alpha}\partial_{s_\beta}\Phi_{\boldsymbol{\omega}}(\boldsymbol{s})-\partial_{s_\alpha}\partial_{s_\beta}\Phi_{\tilde{\boldsymbol{\omega}}}(\boldsymbol{s})|\leq 18H^{3D+1}D^{5/2}\cdot R_\omega^{3D}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2.$$

As a direct result of Lemmas A.1 to A.3, we have

Lemma A.4. *Let $e_i(\cdot;\boldsymbol{\omega}),e_i(\cdot;\tilde{\boldsymbol{\omega}})\in\mathcal{E}_i^{\text{sub}}$ for $i=1,2$. Then for all $\boldsymbol{s}\in\mathcal{D}$, they satisfy both boundedness and Lipschitz conditions with respect to parameters:*

$$\left|e_i(\boldsymbol{s};\boldsymbol{\omega})\right|\leq B_i,\quad \left|e_i(\boldsymbol{s};\boldsymbol{\omega})-e_i(\boldsymbol{s};\tilde{\boldsymbol{\omega}})\right|\leq L_i\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2,$$

where the constants are given by:

$$B_1=3d^2R_0DH^{2D-2}\cdot R_\omega^{2D},\qquad B_2=(H+1)\cdot R_\omega,$$
$$L_1=22d^2R_0D^{5/2}H^{3D+1}\cdot R_\omega^{3D},\quad L_2=2H^D\sqrt{D}\cdot R_\omega^{D-1}.$$

B Proof of Lemma A.3

Note that the activation function $\sigma=\tanh$, which is 1-Lipschitz and has a 1-Lipschitz continuous gradient. For $\ell=1,2,\cdots,D$,

$$\Phi_i^{[\ell]}=\sigma\Big(\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\Big).$$

By Lemma A.2, we have

$$\big|\partial_{s_\alpha}\Phi_i^{[\ell]}\big|=\left|\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\partial_{s_\alpha}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right|\le H^{\ell-1}R_{\boldsymbol{\omega}}^{\ell}.$$

Then, it holds that

$$\begin{aligned}
\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_i^{[\ell]}\big|=&\left|\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right.\\
&\left.+\left[\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\partial_{s_\alpha}\Phi_j^{[\ell-1]}\right]\left[\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\partial_{s_\beta}\Phi_j^{[\ell-1]}\right]\sigma''\left(\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right|\\
\le& R_{\boldsymbol{\omega}}\sum_{j=1}^{H_{\ell-1}}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}+\left[R_{\boldsymbol{\omega}}\sum_{j=1}^{H_{\ell-1}}\partial_{s_\alpha}\Phi_j^{[\ell-1]}\right]^2\\
=& R_{\boldsymbol{\omega}}\sum_{j=1}^{H_{\ell-1}}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}+R_{\boldsymbol{\omega}}^{2\ell}H^{2\ell-2}\\
\le&\cdots\le R_{\boldsymbol{\omega}}^{\ell+1}H^{\ell-1}+R_{\boldsymbol{\omega}}^{\ell+2}H^{\ell}+\cdots+R_{\boldsymbol{\omega}}^{2\ell-1}H^{2\ell-3}+R_{\boldsymbol{\omega}}^{2\ell}H^{2\ell-2}\\
\le&\ell R_{\boldsymbol{\omega}}^{2\ell}H^{2\ell-2}.
\end{aligned}$$

Note that $H_0=d$, $H_D=1$. For $\ell=1$, we have

$$\begin{aligned}
&\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_i^{[1]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_i^{[1]}\big|\\
=&\left|w_{i\alpha}^{[0]}w_{i\beta}^{[0]}\sigma''\left(\sum_{j=1}^{H_0} w_{ij}^{[0]}s_j+c_i^{[0]}\right)-\tilde{w}_{i\alpha}^{[0]}\tilde{w}_{i\beta}^{[0]}\sigma''\left(\sum_{j=1}^{H_0}\tilde{w}_{ij}^{[0]}s_j+\tilde{c}_i^{[0]}\right)\right|\\
\le&\big|w_{i\alpha}^{[0]}-\tilde{w}_{i\alpha}^{[0]}\big|\big|\tilde{w}_{i\beta}^{[0]}\big|\left|\sigma''\left(\sum_{j=1}^{H_0} w_{ij}^{[0]}s_j+c_i^{[0]}\right)\right|\\
&+\big|w_{i\beta}^{[0]}-\tilde{w}_{i\beta}^{[0]}\big|\big|w_{i\alpha}^{[0]}\big|\left|\sigma''\left(\sum_{j=1}^{H_0} w_{ij}^{[0]}s_j+c_i^{[0]}\right)\right|\\
&+\big|\tilde{w}_{i\alpha}^{[0]}\big|\big|\tilde{w}_{i\beta}^{[0]}\big|\left|\sigma''\left(\sum_{j=1}^{H_0} w_{ij}^{[0]}s_j+c_i^{[0]}\right)-\sigma''\left(\sum_{j=1}^{H_0}\tilde{w}_{ij}^{[0]}s_j+\tilde{c}_i^{[0]}\right)\right|
\end{aligned}$$

$$
\begin{aligned}
&\le \left|w_{i\alpha}^{[0]}-\tilde{w}_{i\alpha}^{[0]}\right|R_{\boldsymbol{\omega}}+\left|w_{i\beta}^{[0]}-\tilde{w}_{i\beta}^{[0]}\right|R_{\boldsymbol{\omega}}\\
&\qquad+\left|\sigma''\left(\sum_{j=1}^{H_0}w_{ij}^{[0]}s_j+c_i^{[0]}\right)-\sigma''\left(\sum_{j=1}^{H_0}\tilde{w}_{ij}^{[0]}s_j+\tilde{c}_i^{[0]}\right)\right|R_{\boldsymbol{\omega}}^2\\
&\le (2R_{\boldsymbol{\omega}}+2R_{\boldsymbol{\omega}}^2)\sum_{k=1}^{n_1}\left|\omega_k-\tilde{\omega}_k\right|,
\end{aligned}
$$

where we use $\{w_k\}_{k=1}^{n_1}$ to denote all the parameters $w_{ij}^{[0]}$, $c_i^{[0]}$. For $2\le\ell\le D$, it holds that

$$
\begin{aligned}
&\left|\partial_{s_\alpha}\partial_{s_\beta}\Phi_i^{[\ell]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_i^{[\ell]}\right|\\
\le&\left|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right.\\
&\left.-\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right|\\
&+\left|\left[\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\Phi_j^{[\ell-1]}\right]\left[\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\beta}\Phi_j^{[\ell-1]}\right]\sigma''\left(\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right.\\
&\left.-\left[\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\alpha}\tilde{\Phi}_j^{[\ell-1]}\right]\left[\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\right]\sigma''\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right|.
\end{aligned}
$$

For the first term, we have

$$
\begin{aligned}
&\left|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right.\\
&\qquad\left.-\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right|\\
\le&\left|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\right)\right.\\
&\qquad\left.-\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right|\\
&+\left|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right.\\
&\qquad\left.-\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\sigma'\left(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\right)\right|
\end{aligned}
$$

$$
\begin{aligned}
&+\Big|\sum_{j=1}^{H_{\ell-1}} w_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\sigma'\Big(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\Big)\\
&\quad-\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\sigma'\Big(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\Big)\Big|\\
\le&(\ell-1)R_{\boldsymbol{\omega}}^{2\ell-1}H^{2\ell-3}\Big(\Big|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}-\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\Big|+\big|c_i^{[\ell-1]}-\tilde{c}_i^{[\ell-1]}\big|\Big)\\
&+(\ell-1)R_{\boldsymbol{\omega}}^{2\ell}H^{2\ell-2}\max_j|\Phi_j^{[\ell-1]}-\tilde{\Phi}_j^{[\ell-1]}|+R_{\boldsymbol{\omega}}H\cdot\max_j\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\big|\\
&+\Big|\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}-\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\Big|\cdot\max_j\big|\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\big|\\
\le&(\ell-1)\big(R_{\boldsymbol{\omega}}^{2\ell-1}H^{2\ell-3}+R_{\boldsymbol{\omega}}^{3\ell-2}H^{3\ell-4}+R_{\boldsymbol{\omega}}^{2\ell-2}H^{2\ell-4}\big)\sum_{k=1}^{n_{\ell-1}}\big|\omega_k-\tilde{\omega}_k\big|\\
&+R_{\boldsymbol{\omega}}H\cdot\max_j\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\big|,
\end{aligned}
$$

where in the final step above, we apply the natural extension of Lemma A.1 to the intermediate layer, and we use $\{w_k\}_{k=1}^{n_{\ell-1}}$ to denote all the parameters from the first $\ell-1$ layers. Similarly,

$$
\begin{aligned}
&\Big|\Big[\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\alpha}\Phi_j^{[\ell-1]}\Big]\Big[\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\partial_{s_\beta}\Phi_j^{[\ell-1]}\Big]\sigma''\Big(\sum_{j=1}^{H_{\ell-1}}w_{ij}^{[\ell-1]}\Phi_j^{[\ell-1]}+c_i^{[\ell-1]}\Big)\\
&\quad-\Big[\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\alpha}\tilde{\Phi}_j^{[\ell-1]}\Big]\Big[\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\Big]\sigma''\Big(\sum_{j=1}^{H_{\ell-1}}\tilde{w}_{ij}^{[\ell-1]}\tilde{\Phi}_j^{[\ell-1]}+\tilde{c}_i^{[\ell-1]}\Big)\Big|\\
&\le(R_{\boldsymbol{\omega}}^{2\ell}H^{2\ell-2}+R_{\boldsymbol{\omega}}^{3\ell-1}H^{3\ell-3}+2\ell R_{\boldsymbol{\omega}}^{3\ell-1}H^{3\ell-4}+2R_{\boldsymbol{\omega}}^{2\ell-1}H^{2\ell-3})\sum_{k=1}^{n_{\ell-1}}\big|\omega_k-\tilde{\omega}_k\big|\\
&\le 9\ell H^{3\ell}R_{\boldsymbol{\omega}}^{3\ell}\sum_{k=1}^{n_{\ell-1}}\big|\omega_k-\tilde{\omega}_k\big|,
\end{aligned}
$$

then, it holds that

$$
\begin{aligned}
\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_i^{[\ell]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_i^{[\ell]}\big|\le&R_{\boldsymbol{\omega}}H\cdot\max_j\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_j^{[\ell-1]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_j^{[\ell-1]}\big|\\
&+9\ell H^{3\ell}R_{\boldsymbol{\omega}}^{3\ell}\sum_{k=1}^{n_{\ell-1}}\big|\omega_k-\tilde{\omega}_k\big|\le\cdots\le 9\ell^2H^{3\ell}R_{\boldsymbol{\omega}}^{3\ell}\sum_{k=1}^{n_\ell}\big|\omega_k-\tilde{\omega}_k\big|.
\end{aligned}
$$

And by $\sum_{k=1}^{n_D}\big|\omega_k-\tilde{\omega}_k\big|\le\sqrt{H(H+1)D}\big\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\big\|_2$,

$$
\big|\partial_{s_\alpha}\partial_{s_\beta}\Phi_i^{[D]}-\partial_{s_\alpha}\partial_{s_\beta}\tilde{\Phi}_i^{[D]}\big|\le 18H^{3D+1}\cdot R_{\boldsymbol{\omega}}^{3D}D^2\sqrt{D}\big\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\big\|_2,\quad\forall\boldsymbol{s}\in\mathcal{D}.
$$

C Proof of Lemma 3.1

Note that $\boldsymbol{y}^* \in \mathrm{Y}$. For any $i \in [N]$, by convexity of $\boldsymbol{y} \mapsto F_i(\boldsymbol{x}_t, \boldsymbol{y})$, and (3.6), we have

$$\begin{aligned}
&F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}_t) - F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}^*) \le \left\langle \nabla_{\boldsymbol{y}} F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{y}_t - \boldsymbol{y}^* \right\rangle \\
=& \frac{1}{2\lambda} \cdot 2 \cdot \left\langle \lambda \nabla_{\boldsymbol{y}} F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}_t), \boldsymbol{y}_t - \boldsymbol{y}^* \right\rangle \\
=& \frac{1}{2\lambda} \left[- \left\| \boldsymbol{y}_t - \lambda \nabla_{\boldsymbol{y}} F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}_t) - \boldsymbol{y}^* \right\|_2^2 + \left\| \boldsymbol{y}_t - \boldsymbol{y}^* \right\|_2^2 + \left\| \lambda \nabla_{\boldsymbol{y}} F_{i_{t+1}}(\boldsymbol{x}_t, \boldsymbol{y}_t) \right\|_2^2 \right] \\
\le& \frac{1}{2\lambda} \left[- \left\| \boldsymbol{y}_{t+1} - \boldsymbol{y}^* \right\|_2^2 + \left\| \boldsymbol{y}_t - \boldsymbol{y}^* \right\|_2^2 + \lambda^2 V^2 \right], \quad \forall t = 0, \cdots, T-1.
\end{aligned}$$

Taking conditional expectation $\mathbb{E}[\cdot \,|\, i_1, \cdots, i_t]$ on both sides, we get

$$F(\boldsymbol{x}_t, \boldsymbol{y}_t) - F(\boldsymbol{x}_t, \boldsymbol{y}^*) \le \frac{1}{2\lambda} \left\{ \left\| \boldsymbol{y}_t - \boldsymbol{y}^* \right\|_2^2 - \mathbb{E}\left[\left\| \boldsymbol{y}_{t+1} - \boldsymbol{y}^* \right\|_2^2 \,|\, i_1, \cdots, i_t \right] + \lambda^2 V^2 \right\}.$$

By further taking the full expectation, we obtain

$$\mathbb{E}\left[F(\boldsymbol{x}_t, \boldsymbol{y}_t) \right] - \mathbb{E}\left[F(\boldsymbol{x}_t, \boldsymbol{y}^*) \right] \le \frac{1}{2\lambda} \left\{ \mathbb{E}\left[\left\| \boldsymbol{y}_t - \boldsymbol{y}^* \right\|_2^2 \right] - \mathbb{E}\left[\left\| \boldsymbol{y}_{t+1} - \boldsymbol{y}^* \right\|_2^2 \right] + \lambda^2 V^2 \right\}.$$

Since $\lambda T = 1$, this implies

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

Using above results, we have

$$\begin{aligned}
&\min_{t=0,\cdots,T} \mathbb{E}\left[F(\boldsymbol{x}_t, \boldsymbol{y}_t) \right] \\
\le& \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[F(\boldsymbol{x}_t, \boldsymbol{y}^*) \right] + \frac{\left\| \boldsymbol{y}^* - \boldsymbol{y}_0 \right\|^2}{2} + \frac{V^2}{2T} \\
\le& F(\boldsymbol{x}_0, \boldsymbol{y}^*) + \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\left| F(\boldsymbol{x}_t, \boldsymbol{y}^*) - F(\boldsymbol{x}_0, \boldsymbol{y}^*) \right| \right] + \frac{\left\| \boldsymbol{y}^* - \boldsymbol{y}_0 \right\|_2^2}{2} + \frac{V^2}{2T}.
\end{aligned}$$

Finally, by (3.7) we get

$$\min_{t=0,\cdots,T} \mathbb{E}\left[F(\boldsymbol{x}_t, \boldsymbol{y}_t) \right] \le F(\boldsymbol{x}_0, \boldsymbol{y}^*) + U \left\| \boldsymbol{y}^* \right\|_2 \operatorname{diam}(\mathrm{X}) + \frac{\left\| \boldsymbol{y}^* - \boldsymbol{y}_0 \right\|_2^2}{2} + \frac{V^2}{2T}.$$

D Proof of Theorem 3.2

The proof follows the constructive approach detailed in Theorem 3.2 of [28]. We establish an approximation bound in the $C^2(\bar{\mathcal{D}})$ norm for a target function $v_* \in C^m(\bar{\mathcal{D}})$.

Step 1: Preliminary Lemmas.

We first state two preliminary lemmas that are essential for the error analysis in the $W^{2,\infty}$ space. The first lemma establishes that shallow neural networks can approximate basic monomials in the $W^{2,\infty}$ norm, see Proposition 4.7 in [18].

Lemma D.1. *Let $M>0$, and let $\sigma(x)=\tanh(x)$. Suppose $x_0\in\mathbb{R}$ is a point such that $\sigma^{(r)}(x_0)\neq 0$ for $r\in\{1,2,3\}$. For any $\epsilon\in(0,1)$, there exist $\Phi_{\omega_1}\in\mathcal{F}(1,2,C_1(M)\epsilon^{-1})$ and $\Phi_{\omega_2}\in\mathcal{F}(2,2,C_2(M)\epsilon^{-1})$ such that*

$$\|x-\Phi_{\omega_1}(x)\|_{W^{2,\infty}([-M,M])}\leq\epsilon \quad \text{and} \quad \|x^2-\Phi_{\omega_2}(x)\|_{W^{2,\infty}([-M,M])}\leq\epsilon.$$

The second lemma provides necessary calculus rules for functions in the $W^{2,\infty}$ space.

Lemma D.2. *Let $d_1,d_2\in\mathbb{N}$, and let $\mathcal{D}_1\subset\mathbb{R}^{d_1}$, $\mathcal{D}_2\subset\mathbb{R}^{d_2}$ 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 $\boldsymbol{f}\in W^{2,\infty}(\mathcal{D}_1;\mathbb{R}^{d_2})$ and $g\in W^{2,\infty}(\mathcal{D}_2)$ be Lipschitz functions with $\text{range}(\boldsymbol{f})\subset\mathcal{D}_2$. Then $g\circ\boldsymbol{f}\in W^{2,\infty}(\mathcal{D}_1)$, and*

$$\|g\circ\boldsymbol{f}\|_{W^{2,\infty}(\mathcal{D}_1)}\leq C_1\max\left\{\|g\|_{W^{1,\infty}(\mathcal{D}_2)}\|\boldsymbol{f}\|_{W^{2,\infty}(\mathcal{D}_1;\mathbb{R}^{d_2})},\|g\|_{W^{2,\infty}(\mathcal{D}_2)}\cdot\|\boldsymbol{f}\|^2_{W^{1,\infty}(\mathcal{D}_1;\mathbb{R}^{d_2})}\right\}. \tag{D.1}$$

2. **(Product rule).** *Let $u,v\in W^{2,\infty}(\mathcal{D}_1)$. Then $uv\in W^{2,\infty}(\mathcal{D}_1)$, and*

$$\|uv\|_{W^{2,\infty}(\mathcal{D}_1)}\leq C_2\|u\|_{W^{2,\infty}(\mathcal{D}_1)}\|v\|_{W^{2,\infty}(\mathcal{D}_1)}. \tag{D.2}$$

Proof of the Chain Rule. Let $h(\boldsymbol{x})=g(\boldsymbol{f}(\boldsymbol{x}))$. The zeroth-order term is bounded by $\|h\|_{L^\infty(\mathcal{D}_1)}\leq\|g\|_{L^\infty(\mathcal{D}_2)}$. The first-order derivatives, given by $\partial_{x_i}h=\sum_j((\partial_{y_j}g)\circ\boldsymbol{f})\cdot(\partial_{x_i}f_j)$, are bounded by $\|\partial_{x_i}h\|_{L^\infty}\leq d_2\|g\|_{W^{1,\infty}(\mathcal{D}_2)}\|\boldsymbol{f}\|_{W^{1,\infty}(\mathcal{D}_1)}$. Further, we have

$$\partial_{x_k}\partial_{x_i}h=\sum_{j=1}^{d_2}\left[\left(\sum_{l=1}^{d_2}(\partial_{y_l}\partial_{y_j}g)\circ\boldsymbol{f}\cdot\partial_{x_k}f_l\right)\cdot(\partial_{x_i}f_j)\right]+\sum_{j=1}^{d_2}\left[((\partial_{y_j}g)\circ\boldsymbol{f})\cdot(\partial_{x_k}\partial_{x_i}f_j)\right],$$

which leads to

$$\begin{aligned}\|\partial_{x_k}\partial_{x_i}h\|_{L^\infty}&\leq\sum_{j,l}\|\partial_{y_l}\partial_{y_j}g\|_{L^\infty(\mathcal{D}_2)}\|\partial_{x_k}f_l\|_{L^\infty(\mathcal{D}_1)}\|\partial_{x_i}f_j\|_{L^\infty(\mathcal{D}_1)}\\&\quad+\sum_j\|\partial_{y_j}g\|_{L^\infty(\mathcal{D}_2)}\|\partial_{x_k}\partial_{x_i}f_j\|_{L^\infty(\mathcal{D}_1)}\\&\leq d_2^2\|g\|_{W^{2,\infty}(\mathcal{D}_2)}\|\boldsymbol{f}\|^2_{W^{1,\infty}(\mathcal{D}_1)}+d_2\|g\|_{W^{1,\infty}(\mathcal{D}_2)}\|\boldsymbol{f}\|_{W^{2,\infty}(\mathcal{D}_1)}.\end{aligned}$$

Note that the bound for the second derivative dominates the lower-order terms. This proves the chain rule equation D.1.

Proof of the Product Rule. Let $h(\boldsymbol{x})=u(\boldsymbol{x})v(\boldsymbol{x})$. Directly, we have

$$\|h\|_{L^\infty}\leq\|u\|_{L^\infty}\|v\|_{L^\infty}\leq\|u\|_{W^{2,\infty}(\mathcal{D}_1)}\|v\|_{W^{2,\infty}(\mathcal{D}_1)}.$$

For the first-order derivatives, $\partial_i h=(\partial_i u)v+u(\partial_i v)$, it holds that

$$\|\partial_i h\|_{L^\infty}\leq\|\partial_i u\|_{L^\infty}\|v\|_{L^\infty}+\|u\|_{L^\infty}\|\partial_i v\|_{L^\infty}\leq 2\|u\|_{W^{2,\infty}(\mathcal{D}_1)}\|v\|_{W^{2,\infty}(\mathcal{D}_1)}.$$

For the second-order derivatives, $\partial_j\partial_i h=(\partial_j\partial_i u)v+(\partial_i u)(\partial_j v)+(\partial_j u)(\partial_i v)+u(\partial_j\partial_i v)$. Then, we have

$$\|\partial_j\partial_i h\|_{L^\infty}\leq\|\partial_j\partial_i u\|_{L^\infty}\|v\|_{L^\infty}+\cdots+\|u\|_{L^\infty}\|\partial_j\partial_i v\|_{L^\infty}\leq 4\|u\|_{W^{2,\infty}(\mathcal{D}_1)}\|v\|_{W^{2,\infty}(\mathcal{D}_1)}.$$

Since $\|uv\|_{W^{2,\infty}(\mathcal{D}_1)}=\max_{\|\boldsymbol{\alpha}\|_1\leq 2}\|\partial^{\boldsymbol{\alpha}}(uv)\|_{L^\infty}$, taking the maximum over all derivative orders allows us to choose $C_2=4$ to satisfy the inequality equation (D.2).

Step 2: Local Polynomial Approximation.

We first approximate the target function v_* with a function v_N composed of localized polynomials, in direct analogy to Proposition 4.1 of [18]. The function v_N has the form:

$$v_N:=\sum_{\boldsymbol{m}\in\{0,\cdots,N\}^d}\sum_{\|\boldsymbol{\alpha}\|_1\leq m-1}c_{v_*,\boldsymbol{m},\boldsymbol{\alpha}}\Psi_{\boldsymbol{m}}^s\boldsymbol{x}^{\boldsymbol{\alpha}},$$

where the functions $\Psi_{\boldsymbol{m}}^s$ constitute an tanh-related exponential partition of unity as specified in Lemma 1 of [28]. Also, we have

$$\|v_*-v_N\|_{W^{2,\infty}(\mathcal{D})}\leq C\|v_*\|_{C^m(\bar{\mathcal{D}})}\cdot N^{-(m-2-2\iota)}$$

for a sufficiently large N and $s=N^\iota$, where $\iota>0$ is an arbitrarily small constant. The coefficients are bounded by $|c_{v_*,\boldsymbol{m},\boldsymbol{\alpha}}|\leq C\|v_*\|_{C^m(\bar{\mathcal{D}})}$.

Step 3: Neural Network Approximation of Localized Polynomials.

With the help of Lemmas D.1 and D.2, the construction of sub-networks to approximate $\Psi_{\boldsymbol{m}}^s(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}}$ and its error analysis can be established directly following the methodology of Lemmas 4-6 in [28]. While the network architecture remains unchanged, the error analysis must be adapted to the $W^{2,\infty}$ norm, which introduces a factor $(sN)^2$ in the error bound attributable to the second-order derivative terms. This result is formalized in the following lemma:

Lemma D.3. *Let $d,N,s\geq 1$, $m\geq 2$. For any $0<\epsilon_{NN}<\epsilon^*$, where $\epsilon^*>0$ is sufficiently small, there exists a neural network $\Phi_{\boldsymbol{\omega}}^{m,\boldsymbol{\alpha}}\in\mathcal{F}(H,D,R_{\boldsymbol{\omega}})$ with architecture parameters*

$$H=2^{\lceil\log_2(d+\|\boldsymbol{\alpha}\|_1)\rceil+1},\ D=\lceil\log_2(d+\|\boldsymbol{\alpha}\|_1)\rceil+2,\ R_{\boldsymbol{\omega}}=\max\{3Ns,(3d+3/2)s,C(d)\epsilon_{NN}^{-2}\},$$

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

$$\|\Psi_{\boldsymbol{m}}^s\boldsymbol{x}^{\boldsymbol{\alpha}}-\Phi_{\boldsymbol{\omega}}^{m,\boldsymbol{\alpha}}(\boldsymbol{x})\|_{W^{2,\infty}(\mathcal{D})}\leq C(m,d)(sN)^2\epsilon_{NN},$$

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

Following Theorem 6 of [28], we construct the final parallel network by combining these sub-networks with their corresponding coefficients $c_{v_*,\boldsymbol{m},\boldsymbol{\alpha}}$, as shown blow:

Lemma D.4. *For any $0<\epsilon_{NN}<\epsilon^*$, where $\epsilon^*>0$ is sufficiently small, there exists a parallel neural network $v_{\bar{q},\bar{\omega}}\in\mathcal{PF}_{\bar{q},\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}})$ with parameters*

$$\begin{aligned}&\bar{q}=C_1(m,d)(N+1)^d,\quad \bar{K}=C_2(m,d)(N+1)^d,\quad \bar{H}=2^{\lceil\log_2(d+m-1)\rceil+1},\\&\bar{D}=\lceil\log_2(d+m-1)\rceil+2,\quad R_{\bar{\omega}}=\max\{3Ns,(3d+3/2)s,C(d)\epsilon_{NN}^{-2}\},\end{aligned}$$

such that

$$\|v_N-v_{\bar{q},\bar{\omega}}\|_{W^{2,\infty}(\mathcal{D})}\leq C(m,d)(N+1)^d(sN)^2\epsilon_{NN}.$$

Step 4: Final Error Bound.

To achieve a total approximation error $\|v_*-v_{\bar{q},\bar{\omega}}\|_{C^2(\bar{\mathcal{D}})}\leq\epsilon$, we first have

$$\|v_*-v_{\bar{q},\bar{\omega}}\|_{C^2(\bar{\mathcal{D}})}\leq\|v_*-v_N\|_{W^{2,\infty}(\mathcal{D})}+\|v_N-v_{\bar{q},\bar{\omega}}\|_{W^{2,\infty}(\mathcal{D})}.$$

Then, from Step 1, to ensure $\|v_*-v_N\|_{W^{2,\infty}(\mathcal{D})}\leq\epsilon/2$, we choose N such that $N^{-(m-2-2\iota)}=\mathcal{O}(\epsilon)$. This gives:

$$N=\mathcal{O}\left(\epsilon^{-\frac{1}{m-2-2\iota}}\right)\quad\text{and}\quad s=N^{\iota}=\mathcal{O}\left(\epsilon^{-\frac{\iota}{m-2-2\iota}}\right).$$

From Step 2, to ensure $\|v_N-v_{\bar{q},\bar{\omega}}\|_{W^{2,\infty}(\mathcal{D})}\leq\epsilon/2$, we need to choose the sub-network accuracy ϵ_{NN} such that $C(m,d)(N+1)^d(sN)^2\epsilon_{NN}\leq\epsilon/2$, which yields:

$$\epsilon_{NN}=\mathcal{O}\left(\frac{\epsilon}{s^2N^{d+2}}\right)=\mathcal{O}\left(\epsilon^{1+\frac{d+2+2\iota}{m-2-2\iota}}\right).$$

These choices directly determine the complexity of the resulting parallel network $v_{\bar{q},\bar{\omega}}$, as summarized in the following result which is exactly Theorem 3.2 in the main text:

Theorem D.1. *Given any $v_*\in C^m(\bar{\mathcal{D}})$ with $m\geq3$, for some small $\epsilon^*>0$ and any $0<\epsilon<\epsilon^*$, there exists $v_{\bar{q},\bar{\omega}}\in\mathcal{PF}_{\bar{q},\bar{K}}(\bar{H},\bar{D},R_{\bar{\omega}})$ with*

$$\begin{aligned}&\bar{q}=\lceil C_1\cdot\epsilon^{-\frac{d}{m-2\iota-2}}\rceil,\quad \bar{K}=C_2\cdot\epsilon^{-\frac{d}{m-2\iota-2}},\quad \bar{H}=2^{\lceil\log_2(d+m-1)\rceil+1},\\&\bar{D}=\lceil\log_2(d+m-1)\rceil+2,\quad R_{\bar{\omega}}=C_3\cdot\epsilon^{-\frac{2d+2m}{m-2\iota-2}},\end{aligned}$$

such that

$$\|v_{\bar{q},\bar{\omega}}-v_*\|_{C^2(\bar{\mathcal{D}})}\leq\epsilon,$$

where $0<\iota<1$; Constants C_1, C_2, and C_3 depend exclusively on d and m.

References

[1] 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.

[2] 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.

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

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

[5] 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, 2019.

[6] 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.

[7] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. Does data interpolation contradict statistical optimality? In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1611–1619. PMLR, 2019.

[8] 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.

[9] Susanne Brenner and Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15. Springer Science & Business Media, 2007.

[10] Jingrun Chen, Xurong Chi, Zhouwang Yang, et al. Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method. *Journal of Machine Learning*, 1:268–298, 2022.

[11] 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.

[12] Philippe G Ciarlet. *The Finite Element Method for Elliptic Problems*. SIAM, 2002.

[13] Yongcheng Dai, Bangti Jin, Ramesh Chandra Sau, and Zhi Zhou. Solving elliptic optimal control problems via neural networks and optimality system. *Advances in Computational Mathematics*, 51(4):31, 2025.

[14] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *Proceedings of the 25th international Conference on Machine Learning*, pages 272–279, 2008.

[15] R.M Dudley. The sizes of compact subsets of hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

[16] Weinan E and Bing Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[17] Fengjiang Fu and Xiaoqun Wang. Convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving partial differential equations. *Numerical Mathematics: Theory, Methods and Applications*, 16(3):668–700, 2023.

[18] Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.
[19] 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.
[20] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of Statistics*, 50(2):949, 2022.
[21] 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.
[22] 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.
[23] 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.
[24] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.
[25] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.
[26] Yuling Jiao, Yanming Lai, Dingwei Li, Xiliang Lu, Fengru Wang, Yang Wang, and Jerry Zhijian Yang. 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.
[27] Yuling Jiao, Yanming Lai, Yisu Lo, Yang Wang, and Yunfei Yang. Error analysis of deep Ritz methods for elliptic equations. *Analysis and Applications*, 2023.
[28] Yuling Jiao, Ruoxuan Li, Peiying Wu, Jerry Zhijian Yang, and Pingwen Zhang. DRM revisited: A complete error analysis. *Journal of Machine Learning Research*, 26(115):1–76, 2025.
[29] 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.
[30] 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, 2023.
[31] Yuling Jiao, Jerry Zhijian Yang, Cheng Yuan, and Junyu Zhou. A rate of convergence of weak adversarial neural networks for the second order parabolic PDEs. *Communications in Computational Physics*, 34(3):813–836, 2023.
[32] 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.
[33] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 48(3):1329–1347, 2020.
[34] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-Net: Learning PDEs from data. In *International Conference on Machine Learning*, pages 3208–3216. PMLR, 2018.
[35] 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, 2021.
[36] 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. *ICLR*, 2021.
[37] Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.
[38] 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.
[39] 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.
[40] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
[41] Johannes Müller and Marius Zeinhofer. Error estimates for the variational training of neural networks with boundary penalty. *arXiv preprint arXiv:2103.01007*, 2021.
[42] Preetum Nakkiran, Prayaag Venkat, Sham M Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. In *International Conference on Learning Representations*, 2020.
[43] Guofei Pang, Marta D'Elia, Michael Parks, and George Karniadakis. nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications. *Journal of Computational Physics*, 422:109760, 08 2020.
[44] Guofei Pang, Lu Lu, and George Em Karniadakis. fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing*, 41(4):A2603–A2626, 2019.
[45] 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.
[46] 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.
[47] Justin A. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.
[48] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.
[49] 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.
[50] Yahong Yang and Juncai He. Deeper or wider: A perspective from optimal generalization error with sobolev loss. In *Proceedings of the 41st International Conference on Machine Learning*, pages 56109–56138, 2024.
[51] 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.
[52] 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.