ORIGINAL ARTICLE

Variational based NN and PINN for the Poisson problem

Neela Nataraj[1] · **Ramesh Chandra Sau**[2]

Received: 11 November 2025 / Accepted: 13 November 2025 / Published online: 28 November 2025

Abstract
Recent advances in deep learning have led to efficient solutions of initial and boundary value problems governed by partial differential equations defined in a domain $\Omega \subset \mathbb{R}^d$ for $d \geq 2$. These approaches recast the problem as optimization problems, where the objective is to learn suitable neural network parameters that approximate the solution. In this *expository article*, we focus on neural network-based solvers namely, Deep Ritz Method, Physics-Informed Neural Network, and Variational Physics-Informed Neural Network for the Poisson problem with Dirichlet boundary condition. We discuss a comprehensive error analysis for each of these methods and conduct numerical experiments to assess their computational efficiency.

1 Introduction

Partial differential equations (PDE) model a vast number of natural and manmade phenomena in all areas of science and technology. Explicit solution formulas are only available for very specific types and examples of PDE. Hence, numerical simulations are necessary for most practical applications featuring PDE. A diverse set of methods for approximating PDE numerically is available, such as finite element [10], finite difference [43], finite volume [22], and the more recent methods like the virtual element method [16] and hybrid high order methods [14, 18]. Although these methods are successful in practice, it is still challenging to numerically simulate problems such as uncertainty quantification, multi-scale and multi-physics problems, inverse and constrained optimization problems, PDE in domains with very complex geometries and in very high dimensions.

The recent developments in deep learning have introduced neural networks (NN) as powerful and mesh free function approximators. These networks feature highly nonlinear parameterizations that, when paired with modern hardware accelerators, enable rapid evaluation and differentiation. As universal approximators, they offer the potential to alleviate the curse of dimensionality. The deep learning based method for solving PDE was first introduced in the work by Raissi et al. in 2019 [39]. It is popularly known as physics-informed neural networks (PINN). This method is based on the residual minimization of the PDE and uses the strong form of the PDE. Building on this foundation, a variety of extensions and weak-form approaches have emerged, for example, the Deep Ritz Method (DRM) [20], Variational PINN (VPINN) [6], Weak Adversarial Networks (WAN) [47] as well as operator-learning architectures like DeepONet [33], the Fourier Neural Operator (FNO) [32], and so on. In the article [42], though the authors conduct a comparative study of the neural network-based methods computationally for the elliptic problem, they report the error analysis of only the PINN.

This article is *expository* and discusses the analysis and implementation of a select set of popular approaches for Poisson problem. Though the main focus is on weak formulation based approaches (DRM and VPINN), for the sake of completeness, we also discuss PINN. This article focusses on a unified framework for the error analysis

Communicated by

of *three* popular approaches and does *not* aim to provide an exhaustive survey of all neural network methods for PDE.

Consider the Poisson problem:

$$-\Delta u = f \text{ in } \Omega,\ u = g \text{ on } \partial\Omega, \tag{1.1}$$

defined in a domain $\Omega \subset \mathbb{R}^d$ with $d \geq 2$ and boundary $\partial\Omega$. In this article, we consider three neural network-based PDE solvers, namely, DRM, PINN, and VPINN for (1.1). A schematic diagram of workflow of DRM/PINN/VPINN are described in Figure 1.

The references for the formulations of the methods, loss functions, smoothness of data and solution, and error estimates are listed in Table 1. For the details of notation used in the table, we refer to Section 3, 4, and 5, respectively.

All three methods are characterised by a loss function that is discretised using Monte-Carlo approximation or a quadrature formula as indicated in row 4 of Table 1. The error estimates comprise contributions due to *(a) approximation error* $\mathcal{E}_{app}$, that arises from the approximation of the solution by neural networks, *(b) statistical error* $\mathcal{E}_{stat}$, that arises due to the approximation of the continuous integrals via Monte Carlo methods, *(c) optimization error* $\mathcal{E}_{opt}$, that arises from the approximation of minimizer using a random solver, and *(d) quadrature and interpolation errors (only for VPINN) denoted as* $\mathcal{E}_{quad}$ and $\mathcal{E}_{int}$, *respectively,* see row 6 of Table 1. In the error estimates of all the methods, we avoid the details of discussion of the analysis of the optimization error, which is notoriously challenging and is open to the best of our knowledge. The data smoothness and solution smoothness for each method are described in rows 3 and 5, respectively.

PINN offer a key advantage in high-dimensional or mesh free problems, but in low dimensions, classical numerical methods are typically more efficient as the training cost to optimize the network of PINN is high. In the study by Grossmann et al [26], a comprehensive comparison between PINN and the Finite Element Method (FEM) is conducted through a series of numerical experiments spanning one to three spatial dimensions and it is observed that while PINN can outperform in extrapolation beyond the training domain, they require significantly more training time than FEM. The gap in the practical efficiency of PINN stems from reliance of FEM on fast linear solvers, whereas PINN involve costly nonlinear optimization without comparable efficient solution methods. PINN combine PDE residuals with penalty terms, but often struggle to balance accuracy and constraint enforcement. Wang et al. [44] adjust penalty weights, while Lu et al. [34] enforce boundary conditions exactly. Their reliance

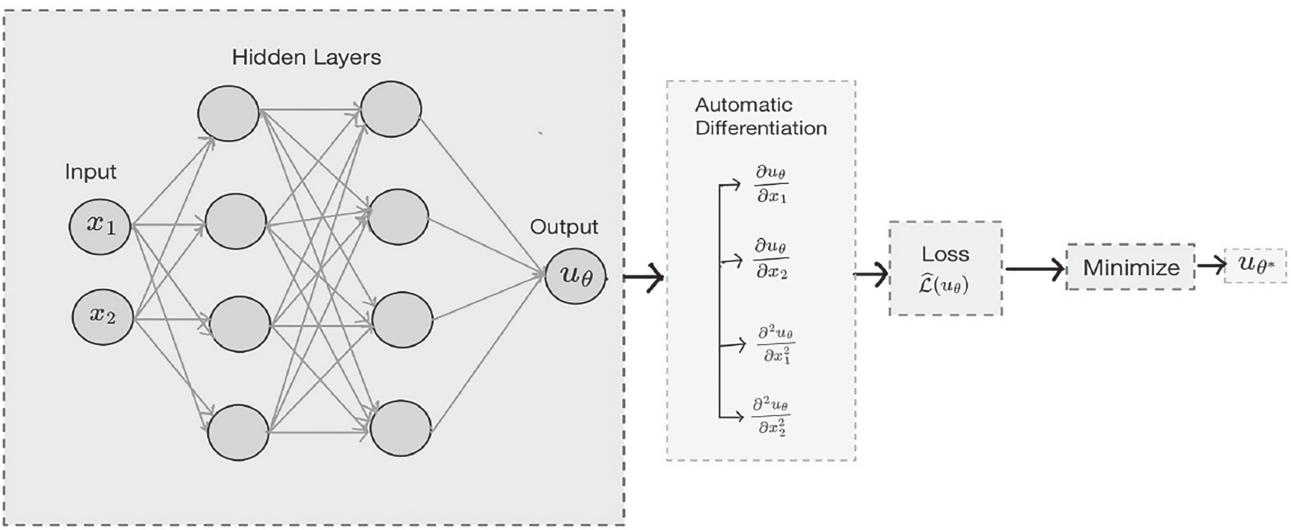


Fig. 1 A diagrammatic illustration of the DRM/PINN/VPINN workflow for Poisson problem

Table 1 Summary of neural network-based methods: DRM, PINN, and VPINN

Category	DRM	PINN	VPINN
1. Reference in this paper	(3.5)	(4.3)	(5.9)
2. Reference in the literature	[20, 29]	[34, 39, 44]	[6, 30]
3. Data smoothness	$f \in C(\bar{\Omega}),\ g \in C(\partial\Omega) \cap H^{1/2}(\partial\Omega)$	$f \in C(\bar{\Omega}),\ g \in C(\partial\Omega)$	$f \in W^{k,\infty}(\Omega),\ g \in H^{k+1/2}(\partial\Omega),\ k \geq 2$
4. Loss function	$\widehat{\mathcal{L}}(u_\theta) = \frac{\lvert\Omega\rvert}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\lvert\nabla u_\theta\rvert^2 - f u_\theta\right)(X_i)$	$\widehat{\mathcal{L}}(u_\theta) = \frac{\lvert\Omega\rvert}{n}\sum_{i=1}^{n}\lvert(\Delta u_\theta + f)(X_i)\rvert^2$	$\widehat{\mathcal{L}}_h(u_\theta) = \sum_{i\in I_h}\left(F_h(\varphi_i) - a_h(\mathcal{I}_h B u_\theta, \varphi_i)\right)^2$
	$+\frac{\lambda}{2}\frac{\lvert\partial\Omega\rvert}{m}\sum_{i=1}^{m}\lvert(u_\theta - g)(Y_i)\rvert^2$	$+\gamma\frac{\lvert\partial\Omega\rvert}{m}\sum_{j=1}^{m}\lvert(u_\theta - g)(Y_j)\rvert^2$	
5. Solution smoothness	$u \in H^2(\Omega)$	$u \in H^3(\Omega)$	$u \in H^{k+1}(\Omega),\ k \geq 2$
6. Error components	$\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\lVert u - u_\theta^{\mathcal{S}}\rVert_{H^1(\Omega)}]$	$\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\lVert u - u_\theta^{\mathcal{S}}\rVert_{L^2(\Omega)}]$	$\lVert u - u_\theta^{\mathcal{S}}\rVert_{H^1(\Omega)}$
	$\lesssim \left(\mathcal{E}_{app} + \mathcal{E}_{stat} + \mathcal{E}_{opt} + \lVert u - u_\lambda\rVert_{H^1(\Omega)}^2\right)^{1/2}$	$\lesssim \left(\mathcal{E}_{app} + \mathcal{E}_{stat} + \mathcal{E}_{opt}\right)^{1/2}$	$\lesssim \mathcal{E}_{int} + \mathcal{E}_{app} + \mathcal{E}_{quad} + \mathcal{E}_{opt}$

on the strong form limits accuracy for solutions with low-regularity. However, the PINN based methods perform well on solving singular perturbation problems in fluid dynamics [38], computational elastodynamics [40], and many other problems in applications.

The DRM, inspired by the classical Ritz method, overcomes this by applying deep learning to the weak/variational problems. In DRM, the traditional trial space is replaced with deep neural networks. By optimizing the network parameters to minimize the Ritz functional, the DRM produces approximations that are accurate and applicable to a wide range of variational problems.

Another weak-form-based approach that is discussed in this article is the VPINN, that aims to approximate weak solutions of PDE. These methods offer flexibility, particularly when dealing with solutions with low regularity or complex domain geometries. VPINN based on the Petrov–Galerkin formulation, use test functions from a polynomial space and represent the trial function with a neural network. Several advancements have been made in VPINN, including FastVPINN [1], a tensor-based approach that reduces computational overhead and efficiently handles complex geometries, as well as adaptive strategies for VPINN, discussed in [5]. Unlike PINN, which yield only L^2 error estimates, DRM, VPINN provide energy norm estimates.

Recent studies have highlighted the significance of the sampling strategy used to evaluate the loss function [23, 46, 48]. This approach is relevant in high-dimensional settings, where traditional quadrature methods become inefficient. Though Monte Carlo methods are often preferred for numerical integration, uniform sampling of collocation points is generally inadequate for capturing localized features of the solution. Adaptive sampling techniques guided by a posteriori error estimators are proposed to enhance accuracy and efficiency [7, 12, 35]. For simplicity of exposition, we stick to uniform sampling in this article.

The remainder of this article is structured as follows. Section 2 introduces preliminaries on neural networks and the notations. A summary of the neural network based methods is presented in Table 1. The DRM and its error analysis is discussed in Section 3. The PINN (resp. VPINN) and its error analysis is discussed in Section 4 (resp. Section 5). Section 6 discusses the algorithm for the three methods and results of numerical experiments that assess the performance of the proposed solvers. The appendix provides classical bounds for approximation and statistical errors.

2 Neural network based PDE solvers

2.1 Deep Neural Networks

Most of the neural network-based solvers for PDE start with approximate solutions by deep neural networks (DNN), which are successive compositions of affine linear transformations and nonlinear activation functions.
Let $L \in \mathbb{N}$ denote the number of layers, and let $\{n_\ell\}_{\ell=0}^{L} \subset \mathbb{N}$ be a sequence of positive integers with $n_0 = d$ and $n_L = 1$. The weight matrix and bias vector associated with the ℓ-th layer are denoted by $W^{(\ell)} = [w_{ij}^{(\ell)}]_{n_\ell \times n_{\ell-1}} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ and $b^{(\ell)} = [b_i^{(\ell)}]_{n_\ell \times 1} \in \mathbb{R}^{n_\ell \times 1}$ respectively. Let $\boldsymbol{n}_i$, for $i = 1, \ldots, L$, denote the total number of nonzero entries in weight matrices and bias vectors in the network up to the i-th layer, so that $\boldsymbol{n}_L$ gives the total number of nonzero entries in weight matrices and bias vectors in the network.

Let $\rho : \mathbb{R} \to \mathbb{R}$ denote the nonlinear activation function. In this article, we use the activation functions: hyperbolic tangent, $\rho(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and sigmoid, $\rho(x) = \frac{1}{1+e^{-x}}$. The choice of ρ depends on the smoothness requirements of the neural network u_θ (defined below), which in turn depends on the order of the PDE.

We denote by $\mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ the class of neural networks with activation function ρ, depth L, total number of nonzero entries in weight matrices and bias vectors in the network $\boldsymbol{n}_L$, and all weights bounded in magnitude by R, more specifically, $\|w_{ij}^{(\ell)}\|_\infty, \|b_i^{(\ell)}\|_\infty \le R$ for all $1 \le i \le n_\ell$, $1 \le j \le n_{\ell-1}$, $1 \le \ell \le L$.
A neural network $u_\theta : \mathbb{R}^d \to \mathbb{R}$ in the class $\mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ parameterized by $\theta \in \mathbb{R}^{\boldsymbol{n}_L}$ is recursively defined as:

$$\begin{cases} u^{(0)}(x) = x, \\ u^{(\ell)}(x) = \rho\big(W^{(\ell)}u^{(\ell-1)} + b^{(\ell)}\big), \quad \ell = 1, \ldots, L-1, \\ u_\theta(x) := u^{(L)}(x) = W^{(L)}u^{(L-1)} + b^{(L)}, \end{cases} \tag{2.1}$$

where $x \in \mathbb{R}^d$. For input $x \in \mathbb{R}^{n_0}$, (2.1) shows u_θ can be written in the following form of successive compositions as

$$u_\theta(x) = u^{(L)} \circ u^{(L-1)} \circ \cdots \circ u^{(2)} \circ u^{(1)} \circ u^{(0)}(x).$$

2.2 Notation

For PINN and DRM, we assume that the domain Ω is a subset of $(-1, 1)^d$ with $d \geq 2$ and has a smooth boundary $\partial\Omega$. For VPINN, $d = 2, 3$ and $\Omega \subset \mathbb{R}^d$. For $1 \leq p \leq \infty$, the L^p norm of a function $u \in L^p(\Omega)$ is defined as $\|u\|_{L^p(\Omega)} := \left(\int_\Omega |u(x)|^p \, dx\right)^{1/p}$ when $1 \leq p < \infty$, and $\|u\|_{L^\infty(\Omega)} = \inf\{B : |u(x)| \leq B \text{ a.e. } x \text{ in } \Omega\}$ when $p = \infty$. For $k \in \mathbb{N} \cup \{0\}$ and $1 \leq p \leq \infty$ the Sobolev space $W^{k,p}(\Omega)$ is defined as $W^{k,p}(\Omega) := \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega) \text{ for all } \alpha \text{ with } |\alpha| \leq k\}$, where the derivatives $D^\alpha u$ are understood in the weak sense. For $1 \leq p < \infty$, the norm on $W^{k,p}(\Omega)$ is defined by $\|u\|_{W^{k,p}(\Omega)} := \left(\sum_{|\alpha|\leq k} \|D^\alpha u\|^p_{L^p(\Omega)}\right)^{1/p}$. For $p = \infty$, $\|u\|_{W^{k,\infty}(\Omega)} := \max_{|\alpha|\leq k} \|D^\alpha u\|_{L^\infty(\Omega)}$. When $p = 2$, the space $W^{k,2}(\Omega)$ is denoted by $H^k(\Omega)$. That is, $H^k(\Omega) = W^{k,2}(\Omega)$. For $0 < s < 1$, the fractional Sobolev space $H^s(\Omega)$ is defined by $H^s(\Omega) := \left\{u \in L^2(\Omega) : \int_\Omega \int_\Omega \frac{|u(x)-u(y)|^2}{|x-y|^{d+2s}} \, dx \, dy < \infty\right\}$, and its norm is defined by $\|u\|_{H^s(\Omega)} := \left(\|u\|^2_{L^2(\Omega)} + \int_\Omega \int_\Omega \frac{|u(x)-u(y)|^2}{|x-y|^{d+2s}} \, dx \, dy\right)^{1/2}$.

3 Deep Ritz method

The DRM [20] draws inspiration from the classical Ritz technique, which uses the minimization of an energy functional over a selected set of trial functions to solve a variational problem. The trial space is chosen as DNN and an optimization of the neural network parameters is then carried out to minimize the associated energy functional.

Assume that the source term $f \in C(\bar{\Omega})$, and the boundary data $g \in C(\partial\Omega) \cap H^{1/2}(\partial\Omega)$. Let $\mathcal{K} := \{u \in H^1(\Omega) : u = g \text{ on } \partial\Omega\}$. The Ritz formulation corresponding to (1.1) seeks $u \in \mathcal{K}$ such that

$$u = \underset{v\in\mathcal{K}}{\operatorname{argmin}} \mathcal{L}(v) := \int_\Omega \left(\frac{1}{2}|\nabla v|^2 - fv\right) dx. \tag{3.1}$$

The boundary condition $u = g$ is weakly imposed in (3.1) and for $\lambda > 0$, the *modified Ritz formulation* for (1.1) reads

$$u_\lambda = \underset{v\in H^1(\Omega)}{\operatorname{argmin}} \mathcal{L}_\lambda(v) := \int_\Omega \left(\frac{1}{2}|\nabla v|^2 - fv\right) dx + \frac{\lambda}{2}\int_{\partial\Omega} |v - g|^2 \, ds. \tag{3.2}$$

Let $U(\Omega)$ and $U(\partial\Omega)$ denote the uniform probability distributions over the domain Ω and its boundary $\partial\Omega$, respectively, with $|\Omega|$ and $|\partial\Omega|$ denoting their Lebesgue measures. Let $\mathbb{E}_{U(\Omega)}$ and $\mathbb{E}_{U(\partial\Omega)}$ denote expectations with respect to these distributions ([41, subsections 2.3.1 & 2.4.2]). The loss functional (3.2) can be represented

in terms of expectations (see [8, Sec 21, page 273]) as:

$$\mathcal{L}_\lambda(v) = |\Omega|\,\mathbb{E}_{X\sim U(\Omega)}\left[\left(\frac{1}{2}|\nabla v|^2 - fv\right)(X)\right] + \frac{\lambda}{2}|\partial\Omega|\,\mathbb{E}_{Y\sim U(\partial\Omega)}\left[|(v-g)(Y)|^2\right]. \tag{3.3}$$

Let $\{X_i\}_{i=1}^n$ and $\{Y_j\}_{j=1}^m$ denote independent and identically distributed (i.i.d.) samples drawn uniformly from Ω and $\partial\Omega$, respectively; that is, $\mathbb{X} = \{X_i\}_{i=1}^n \sim U(\Omega)$ and $\mathbb{Y} = \{Y_j\}_{j=1}^m \sim U(\partial\Omega)$. By replacing v by a neural network $u_\theta \in \mathcal{U} := \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ in (3.3), and applying Monte Carlo integration (see [41, subsection 11.1]), we obtain the empirical loss function $\widehat{\mathcal{L}_\lambda}(u_\theta)$ as

$$\widehat{\mathcal{L}_\lambda}(u_\theta) = \frac{|\Omega|}{n}\sum_{i=1}^n\Big((\frac{1}{2}|\nabla u_\theta|^2 - fu_\theta)(X_i)\Big) + \frac{\lambda}{2}\frac{|\partial\Omega|}{m}\sum_{i=1}^m |(u_\theta - g)(Y_i)|^2. \tag{3.4}$$

Note that the choices of activation functions ρ for this method are hyperbolic tangent and sigmoid. The DRM formulation for (1.1) is the optimization problem that seeks $u_{\theta^*} \in \mathcal{U} := \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ such that

$$u_{\theta^*} \in \underset{u_\theta\in\mathcal{U}}{\operatorname{argmin}} \ \widehat{\mathcal{L}_\lambda}(u_\theta). \tag{3.5}$$

Remark 3.1 (solving (3.5)) The neural network parameter θ satisfies the box constraint , namely $|\theta|_{\ell^\infty} \le R$ for some suitable $R > 0$. Since this ensures that the feasible set is compact in $\mathbb{R}^{\boldsymbol{n}_L}$, the optimization problem defined in (3.5) admits a solution. Moreover, when the activation function ρ is smooth, the empirical loss function $\widehat{\mathcal{L}}(u_\theta)$ is continuous with respect to θ. In practice, approximation of minimizer u_{θ^*} is typically obtained using standard optimization algorithms such as the limited-memory BFGS method [11], Adam optimizer [31], and AdaGrad [19]. All these are widely available in public deep learning frameworks like PyTorch and TensorFlow. Evaluating the loss $\widehat{\mathcal{L}}(u_\theta)$ requires computing spatial derivatives of the neural network output u_θ with respect to the input variable x. Additionally, gradient-based optimizers necessitate computation of the derivative of the loss function with respect to the network parameters θ. Both types of derivatives can be efficiently computed using automatic differentiation techniques [3]; for instance, via `torch.autograd` in PyTorch.

3.1 Error analysis

The Euler-Lagrange formulation that corresponds to (3.2) leads to (3.6) below that seeks u_λ such that

$$-\Delta u_\lambda = f \quad \text{in } \Omega, \quad u_\lambda + \frac{1}{\lambda}\frac{\partial u_\lambda}{\partial n} = g \quad \text{on } \partial\Omega. \tag{3.6}$$

For simplicity of analysis, we choose $\lambda > 1$.

The error analysis follows the arguments in [29] and [37]. We first derive a key estimate in Lemma 3.1 where the error is expressed as a combination of contributions from statistical, approximation, and optimization errors. This is followed by estimates for approximation error in Lemma 3.2 and statistical error in Lemma 3.3. The final error estimate is consolidated in Theorem 3.1.

Lemma 3.1 *(key estimate) Let u (resp. u_λ) solve* (3.1) *(resp.* (3.6)*) and satisfy $u \in H^2(\Omega)$. Let u_{θ^*} be a global minimizer to the loss function $\widehat{\mathcal{L}_\lambda}(u_\theta)$ in* (3.5) *and $u_\theta^{\mathcal{S}}$ be the solution to* (3.5) *generated by a random solver $\mathcal{S}$. Then it holds that*

$$\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\|u - u_{\theta^*}\|_{H^1(\Omega)}] \lesssim \big(\frac{1}{\lambda^2} + \mathcal{E}_{stat} + \lambda\mathcal{E}_{app} + \mathcal{E}_{opt}\big)^{1/2}, \tag{3.7}$$

where $\{X_k\}_{k=1}^n$ *and* $\{Y_k\}_{k=1}^m$ *are interior and boundary collocation points, respectively, the constant absorbed in* $\lesssim$ *depends on the constants from trace inequality, inverse trace inequality,* $\|u\|_{H^2(\Omega)}$*, and the errors* $\mathcal{E}_{app}$*,* $\mathcal{E}_{stat}$*, and* $\mathcal{E}_{opt}$ *are the approximation, statistical, and optimization errors, respectively, defined by*

$$\mathcal{E}_{app} = \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)}, \quad \mathcal{E}_{stat} = \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\sup_{u_\theta \in \mathcal{U}} (\mathcal{L}_\lambda(u_\theta) - \widehat{\mathcal{L}_\lambda}(u_\theta)) + \sup_{u_\theta \in \mathcal{U}} (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta))],$$

$$\mathcal{E}_{opt} = \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\widehat{\mathcal{L}_\lambda}(u_\theta^S) - \widehat{\mathcal{L}_\lambda}(u_{\theta^*}) + \left\|u_\theta^S - u_{\theta^*}\right\|^2_{H^1(\Omega)}].$$

Remark 3.2 It is important to note that the expectations with respect to the random collocation points $\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}\}$ arise naturally in (3.7) when estimating the statistical error. This aspect is discussed in detail in Appendix B of this article. A similar type of estimate can be found in [29, Theorem 6.1] and [17, Theorem 3.7].

Proof A triangle inequality leads to

$$\|u - u_{\theta^*}\|_{H^1(\Omega)} \leq \|u - u_\lambda\|_{H^1(\Omega)} + \left\|u_\lambda - u_\theta^S\right\|_{H^1(\Omega)} + \left\|u_\theta^S - u_{\theta^*}\right\|_{H^1(\Omega)}. \tag{3.8}$$

The rest of the proof is divided into two steps: *Step 1* deals with the convergence of solution u_λ of the penalized problem to u as $\lambda \to \infty$; that is, the bound for $\|u - u_\lambda\|_{H^1(\Omega)}$ and *Step 2* bounds $\left\|u_\lambda - u_\theta^S\right\|_{H^1(\Omega)}$ in terms of approximation error, statistical error, and part of optimization error. The term $\left\|u_\theta^S - u_{\theta^*}\right\|_{H^1(\Omega)}$ is a part of the optimization error and it is assumed that this term can be controlled.

Step 1: (control of $\|u - u_\lambda\|_{H^1(\Omega)}$).

For all $v \in H^1(\Omega)$, (1.1) leads to the weak form

$$\int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, ds = \int_\Omega f v \, dx. \tag{3.9}$$

An application of (3.9) in (3.2) and elementary algebraic manipulations lead to

$$\mathcal{L}_\lambda(v) = R_\lambda(v) + \int_{\partial\Omega} \frac{\partial u}{\partial n} g \, ds - \frac{1}{2}\|\nabla u\|^2_{L^2(\Omega)} - \frac{1}{2\lambda}\left\|\frac{\partial u}{\partial n}\right\|^2_{L^2(\partial\Omega)} \tag{3.10}$$

with

$$R_\lambda(v) := \frac{1}{2}\|\nabla(v-u)\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\left\|v - g + \frac{1}{\lambda}\frac{\partial u}{\partial n}\right\|^2_{L^2(\partial\Omega)}. \tag{3.11}$$

Since u_λ is the minimiser of (3.2), it is also a minimiser of R_λ. Also, since $u \in H^2(\Omega)$ and $\frac{\partial u}{\partial n} \in H^{\frac{1}{2}}(\partial\Omega)$, there exists $\phi \in H^1(\Omega)$ such that $\phi|_{\partial\Omega} = -\frac{\partial u}{\partial n}$ and $\|\phi\|_{H^1(\Omega)} \lesssim \left\|\frac{\partial u}{\partial n}\right\|_{H^{1/2}(\partial\Omega)}$, where $\lesssim$ depends on the constant from the inverse trace inequality. For $\bar{u} = \frac{1}{\lambda}\phi + u$, we have

$$\frac{1}{2}\|\nabla(u_\lambda - u)\|^2_{L^2(\Omega)} \leq R_\lambda(u_\lambda) \leq R_\lambda(\bar{u}) = \frac{1}{2\lambda^2}\|\phi\|^2_{H^1(\Omega)} \lesssim \frac{1}{2\lambda^2}\left\|\frac{\partial u}{\partial n}\right\|^2_{H^{1/2}(\partial\Omega)} \lesssim \frac{1}{2\lambda^2}\|u\|^2_{H^2(\Omega)} \tag{3.12}$$

with trace theorem applied in the last step. The above displayed estimate also bounds the second term in the definition of $R_\lambda(u_\lambda)$ from (3.11) as $\left\|u_\lambda - g + \frac{1}{\lambda}\frac{\partial u}{\partial n}\right\|_{L^2(\partial\Omega)} \lesssim \frac{1}{\lambda^{3/2}}\|u\|_{H^2(\Omega)}$. This, a triangle inequality, $u = g$ on

$\partial\Omega$, and a trace inequality shows

$$\|u_\lambda - u\|_{L^2(\partial\Omega)} \le \|u_\lambda - g + \lambda^{-1}\partial u/\partial n\|_{L^2(\partial\Omega)} + \|g - \lambda^{-1}\partial u/\partial n - u\|_{L^2(\partial\Omega)} \lesssim \left(\frac{1}{\lambda^{3/2}} + \frac{1}{\lambda}\right)\|u\|_{H^2(\Omega)}. \tag{3.13}$$

Note that the Poincaré-Friedrichs inequality [9, equation (1.1), page 306] leads to the norm equivalence

$$\|w\|_{H^1(\Omega)} \lesssim \|\nabla w\|_{L^2(\Omega)} + \|w\|_{L^2(\partial\Omega)} \lesssim \|w\|_{H^1(\Omega)} \text{ for all } w \in H^1(\Omega). \tag{3.14}$$

A combination of this, (3.12), and (3.13) shows $\|u - u_\lambda\|_{H^1(\Omega)} \lesssim (\frac{1}{\lambda} + \frac{1}{\lambda^{3/2}})$, where the constant absorbed in $\lesssim$ depends on $\|u\|_{H^2(\Omega)}$ and the constants from trace inequalities.

Step 2: (control of $\left\|u_\theta^{\mathcal{S}} - u_\lambda\right\|_{H^1(\Omega)}$). The definition of $\mathcal{L}_\lambda(v)$ from (3.2) for the choice $v := u_\theta^{\mathcal{S}} \in H^1(\Omega)$ shows

$$\mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) = \frac{1}{2}\left\|\nabla u_\theta^{\mathcal{S}}\right\|^2_{L^2(\Omega)} - (f, u_\theta^{\mathcal{S}}) + \frac{\lambda}{2}\left\|u_\theta^{\mathcal{S}} - g\right\|^2_{L^2(\partial\Omega)}.$$

This, the weak form of (3.6) (tested with $u_\theta^{\mathcal{S}}$), and elementary manipulations lead to

$$\begin{aligned}\mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) &= \frac{1}{2}\left\|\nabla u_\theta^{\mathcal{S}}\right\|^2_{L^2(\Omega)} - \int_\Omega \nabla u_\lambda \cdot \nabla u_\theta^{\mathcal{S}}\, dx - \lambda\int_{\partial\Omega} u_\lambda u_\theta^{\mathcal{S}}\, ds + \lambda\int_{\partial\Omega} g u_\theta^{\mathcal{S}}\, ds + \frac{\lambda}{2}\left\|u_\theta^{\mathcal{S}} - g\right\|^2_{L^2(\partial\Omega)} \\ &= \frac{1}{2}\left\|\nabla(u_\theta^{\mathcal{S}} - u_\lambda)\right\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\left\|(u_\theta^{\mathcal{S}} - u_\lambda)\right\|^2_{L^2(\partial\Omega)} - \frac{1}{2}\|\nabla u_\lambda\|^2_{L^2(\Omega)} - \frac{\lambda}{2}\|u_\lambda\|^2_{L^2(\partial\Omega)} + \frac{\lambda}{2}\|g\|^2_{L^2(\partial\Omega)}. \end{aligned} \tag{3.15}$$

Analogously, the definition of $L_\lambda(v)$ in (3.2) with the choice $v := u_\lambda$ and (3.6) (tested with u_λ) yields

$$\mathcal{L}_\lambda(u_\lambda) = -\frac{1}{2}\|\nabla u_\lambda\|^2_{L^2(\Omega)} - \frac{\lambda}{2}\|u_\lambda\|^2_{L^2(\partial\Omega)} + \frac{\lambda}{2}\|g\|^2_{L^2(\partial\Omega)} \tag{3.16}$$

A combination of (3.15) and (3.16) with the norm equivalence from (3.14) leads to

$$\left\|u_\theta^{\mathcal{S}} - u_\lambda\right\|^2_{H^1(\Omega)} \lesssim \frac{1}{2}\left\|\nabla(u_\theta^{\mathcal{S}} - u_\lambda)\right\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\left\|(u_\theta^{\mathcal{S}} - u_\lambda)\right\|^2_{L^2(\partial\Omega)} = \mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) - \mathcal{L}_\lambda(u_\lambda). \tag{3.17}$$

Algebraic manipulations show

$$\begin{aligned}\mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) - \mathcal{L}_\lambda(u_\lambda) = (\mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) - \widehat{\mathcal{L}_\lambda}(u_\theta^{\mathcal{S}})) + (\widehat{\mathcal{L}_\lambda}(u_\theta^{\mathcal{S}}) - \widehat{\mathcal{L}_\lambda}(u_{\theta^*})) + (\widehat{\mathcal{L}_\lambda}(u_{\theta^*}) - \widehat{\mathcal{L}_\lambda}(u_\theta)) \\ + (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta)) + (\mathcal{L}_\lambda(u_\theta) - \mathcal{L}_\lambda(u_\lambda)) \text{ for all } u_\theta \in \mathcal{U}. \end{aligned} \tag{3.18}$$

The last term $\mathcal{L}_\lambda(u_\theta) - \mathcal{L}_\lambda(u_\lambda)$ in (3.18) can be estimated as follows: The second and third expressions in (3.17) (with the choice $u_\theta^S := u_\theta$) yield

$$\mathcal{L}_\lambda(u_\theta) - \mathcal{L}_\lambda(u_\lambda) = \frac{1}{2}\|\nabla(u_\theta - u_\lambda)\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u_\theta - u_\lambda\|^2_{L^2(\partial\Omega)}.$$

An application of the trace inequality in the second term above yields

$$\mathcal{L}_\lambda(u_\theta) - \mathcal{L}_\lambda(u_\lambda) \le \frac{1}{2}\|\nabla(u_\theta - u_\lambda)\|^2_{L^2(\Omega)} + \frac{\lambda}{2}C_{tr}\|u_\theta - u_\lambda\|^2_{H^1(\Omega)}$$

$$\lesssim \lambda \|u_\theta - u_\lambda\|^2_{H^1(\Omega)}. \tag{3.19}$$

The constant absorbed in $\lesssim$ depends on the trace inequality constant. Utilizing (3.19), and the fact that u_{θ^*} is a minimum (that leads to $\widehat{\mathcal{L}_\lambda}(u_{\theta^*}) - \widehat{\mathcal{L}_\lambda}(u_\theta) \leq 0$) in (3.18), we infer

$$\begin{aligned}\mathcal{L}_\lambda(u_\theta^{\mathcal{S}}) - \mathcal{L}_\lambda(u_\lambda) \lesssim & \sup_{u_\theta \in \mathcal{U}} (\mathcal{L}_\lambda(u_\theta) - \widehat{\mathcal{L}_\lambda}(u_\theta)) + (\widehat{\mathcal{L}_\lambda}(u_\theta^{\mathcal{S}}) - \widehat{\mathcal{L}_\lambda}(u_{\theta^*})) \\ & + \sup_{u_\theta \in \mathcal{U}} (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta)) + \lambda \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)}. \end{aligned} \tag{3.20}$$

Substitute (3.20) in (3.17), and take expectations with respect to the collocation points $\{X_k\}_{k=1}^n$ and $\{Y_k\}_{k=1}^m$ to obtain

$$\begin{aligned}\mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\left\|u_\theta^{\mathcal{S}} - u_\lambda\right\|^2_{H^1(\Omega)}] \lesssim & \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\sup_{u_\theta \in \mathcal{U}} (\mathcal{L}_\lambda(u_\theta) - \widehat{\mathcal{L}_\lambda}(u_\theta)) + \sup_{u_\theta \in \mathcal{U}} (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta))] \\ & + \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\widehat{\mathcal{L}_\lambda}(u_\theta^{\mathcal{S}}) - \widehat{\mathcal{L}_\lambda}(u_{\theta^*})] + \lambda \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)}. \end{aligned} \tag{3.21}$$

Perform elementary manipulations in (3.8), take expectations with respect to collocation points $\{X_k\}_{k=1}^n$ and $\{Y_k\}_{k=1}^m$, and utilise (3.21), to infer

$$\begin{aligned}\mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\|u - u_{\theta^*}\|^2_{H^1(\Omega)}] \lesssim & \|u - u_\lambda\|^2_{H^1(\Omega)} + \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\left\|u_\theta^S - u_{\theta^*}\right\|^2_{H^1(\Omega)}] \\ & + \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\sup_{u_\theta \in \mathcal{U}} (\mathcal{L}_\lambda(u_\theta) - \widehat{\mathcal{L}_\lambda}(u_\theta)) + \sup_{u_\theta \in \mathcal{U}} (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta))] \\ & + \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\widehat{\mathcal{L}_\lambda}(u_\theta^{\mathcal{S}}) - \widehat{\mathcal{L}_\lambda}(u_{\theta^*})] + \lambda \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)}. \end{aligned} \tag{3.22}$$

Utilize $\|u - u_\lambda\|_{H^1(\Omega)} \lesssim (\frac{1}{\lambda} + \frac{1}{\lambda^{3/2}})$ (from *Step 1*) and the fact that $(\mathbb{E}[f])^2 \leq \mathbb{E}[f^2]$ (that follows from Cauchy-Schwarz inequality) in the right-hand side and left-hand side of (3.22) respectively, and finally take square root on both the sides to arrive at (3.7). □

Assume that the solution satisfies the regularity assumption [29]

$$\|u_\lambda\|_{H^2(\Omega)} \lesssim \lambda \tag{3.23}$$

with a constant in $\lesssim$ that depends on the data $\|f\|_{L^2(\Omega)}$ and $\|g\|_{H^{1/2}(\partial\Omega)}$. Recall that we assume $\lambda > 1$ for the analysis and it tends to ∞ in the limiting case.

Lemma 3.2 *(control of approximation error) Fix a tolerance $\varepsilon \in (0, 1)$. Let $u_\lambda \in H^2(\Omega)$ be the solution of* (3.2) *that satisfies* (3.23)*. Then there exists a neural network function $u_\theta \in \mathcal{U} = \mathcal{N}_\rho\Big(c\log(d+2), c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}, c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}\Big)$ with ρ as hyperbolic tangent or sigmoid function such that*

$$\mathcal{E}_{app} := \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)} \lesssim \lambda^2 \varepsilon^5,$$

where the constant in $\lesssim$ depends on $\|f\|_{L^2(\Omega)}$, $\|g\|_{H^{1/2}(\partial\Omega)}$, and the parameter $\mu > 0$.

Proof Elementary manipulations show

$$\mathcal{E}_{app} := \inf_{u_\theta \in \mathcal{U}} \|u_\lambda - u_\theta\|^2_{H^1(\Omega)} = \|u_\lambda\|^2_{H^2(\Omega)} \inf_{u_\theta \in \mathcal{U}} \left\| \frac{u_\lambda}{\|u_\lambda\|_{H^2(\Omega)}} - \frac{u_\theta}{\|u_\lambda\|_{H^2(\Omega)}} \right\|^2_{H^1(\Omega)}$$
$$= \|u_\lambda\|^2_{H^2(\Omega)} \inf_{u_\theta \in \mathcal{U}} \left\| \frac{u_\lambda}{\|u_\lambda\|_{H^2(\Omega)}} - u_\theta \right\|^2_{H^1(\Omega)}.$$

Apply the approximation result Lemma A.1 with $m = 2$, $p = 2$, $k = 1$, and $\delta = \varepsilon^{5/2}$. This implies that there exists a neural network function u_θ with number of layers $L = c\log(d+2)$, number of non-zero parameters $\boldsymbol{n}_L = c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}$, and bounds on the parameters $R = c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}$, that is,

$$u_\theta \in \mathcal{N}_\rho\Big(c\log(d+2), c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}, c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}\Big)$$

such that $\left\| \frac{u_\lambda}{\|u_\lambda\|_{H^2(\Omega)}} - u_\theta \right\|_{H^1(\Omega)} \leq \varepsilon^{5/2}$. A combination of all this shows

$$\mathcal{E}_{app} \leq \|u_\lambda\|^2_{H^2(\Omega)}\, \varepsilon^5 \lesssim \lambda^2 \varepsilon^5 \tag{3.24}$$

with (3.23) in the last step. Here the parameter μ is non-negative and the constant in $\lesssim$ depends on $\|f\|_{L^2(\Omega)}$ and $\|g\|_{H^{1/2}(\partial\Omega)}$. □

Lemma 3.3 *(statistical error) A bound for the statistical error in terms of the number of samples n, m, respectively, drawn from Ω and $\partial\Omega$, the number of non-zero entries $\mathbf{n}_L$ in the weight matrices and bias vectors, and the bounds of the weights and biases R is given by*

$$\mathcal{E}_{stat} := \mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\sup_{u_\theta \in \mathcal{U}} (\mathcal{L}_\lambda(u_\theta) - \widehat{\mathcal{L}_\lambda}(u_\theta)) + \sup_{u_\theta \in \mathcal{U}} (\widehat{\mathcal{L}_\lambda}(u_\theta) - \mathcal{L}_\lambda(u_\theta))]$$
$$\lesssim n^{-\frac{1}{4}} \boldsymbol{n}_L^{\frac{7L}{2}-\frac{9}{4}} R^{\frac{7L}{2}} + \frac{\lambda}{2} m^{-\frac{1}{4}} \boldsymbol{n}_L^{\frac{L}{2}+\frac{15}{4}} R^{\frac{L}{2}+\frac{5}{2}},$$

where the constant in $\lesssim$ depends on d, $\|f\|_{L^\infty(\Omega)}$, and $\|g\|_{L^\infty(\partial\Omega)}$.

Proof For the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$ defined by

$$\mathcal{F}_i = \left\{ h_i(u_\theta;\cdot) : \Omega \to \mathbb{R} |\ h_i(u_\theta; x) = \left(\frac{1}{2}|\nabla u_\theta|^2 - f u_\theta\right)(x), u_\theta \in \mathcal{U} \right\} \text{ and}$$
$$\mathcal{F}_b = \{ h_b(u_\theta;\cdot) : \partial\Omega \to \mathbb{R} |\ h_b(u_\theta; x) = (u_\theta - g)^2(x), u_\theta \in \mathcal{U} \},$$

it is enough to determine M_i, M_b, Λ_i, and Λ_b in **(H1)** and **(H2)** given below and utilize them in (B.10) (with $\beta = \lambda/2$) to estimate the statistical error (see Appendix B for details). We show that

$$\textbf{(H1)}: \begin{cases} \|h_i(u_\theta;\cdot)\|_{L^\infty(\Omega)} \leq M_i(d, \boldsymbol{n}_L, R, \|f\|_{L^\infty(\Omega)}) \text{ for all } h_i(u_\theta;\cdot) \in \mathcal{F}_i \\ \|h_b(u_\theta;\cdot)\|_{L^\infty(\partial\Omega)} \leq M_b(d, \boldsymbol{n}_L, R, \|f\|_{L^\infty(\Omega)}) \text{ for all } h_b(u_\theta;\cdot) \in \mathcal{F}_b \end{cases}$$
$$\textbf{(H2)}: \begin{cases} \|h_i(u_\theta;\cdot) - h_i(u_{\tilde\theta};\cdot)\|_{L^\infty(\Omega)} \leq \Lambda_i \|\theta - \tilde\theta\|_{\ell^2} \text{ for all } h_i(u_\theta;\cdot), h_i(u_{\tilde\theta};\cdot) \in \mathcal{F}_i \\ \|h_b(u_\theta;\cdot) - h_b(u_{\tilde\theta};\cdot)\|_{L^\infty(\partial\Omega)} \leq \Lambda_b \|\theta - \tilde\theta\|_{\ell^2} \text{ for all } h_b(u_\theta;\cdot), h_b(u_{\tilde\theta};\cdot) \in \mathcal{F}_b \end{cases}$$

hold with

$$M_i := d\boldsymbol{n}_L^{2(L-1)}R^{2L} + (n_{L-1}+1)R\|f\|_{L^\infty(\Omega)},\ M_b := ((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)})^2$$

and Lipschitz continuity constants

$$\Lambda_i = d^2\boldsymbol{n}_L^{3L-5/2}R^{3L-1}(L+1) + \boldsymbol{n}_L^{L-1/2}R^{L-1}\|f\|_{L^\infty(\Omega)},\ \Lambda_b = 2((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)})\boldsymbol{n}_L^{L-\frac{1}{2}}R^{L-1}.$$

For $h_i(u_\theta;\cdot) \in \mathcal{F}_i$, we have $h_i(u_\theta;x) = \frac{1}{2}\sum_{i=1}^d |\partial_{x_i}u_\theta(x)|^2 - (fu_\theta)(x)$. Utilize (B.4) and (B.6) to infer

$$|h_i(u_\theta;x)| \le \frac{1}{2}d\left(\left(\prod_{k=1}^{L-1}n_k\right)R^L\right)^2 + (n_{L-1}+1)R\|f\|_{L^\infty(\Omega)} \le d\boldsymbol{n}_L^{2(L-1)}R^{2L} + (n_{L-1}+1)R\|f\|_{L^\infty(\Omega)} = M_i.$$

For $h_b(u_\theta;x) \in \mathcal{F}_b$, the bound M_b is straight forward from the definition $h_i(u_\theta;\cdot) := (u_\theta - g)^2$ and (B.4). Elementary manipulations using the definition of h show

$$\begin{aligned} h_i(u_\theta;x) - h_i(u_{\tilde\theta};x) &= \left(\frac{1}{2}|\nabla u_\theta|^2 - fu_\theta\right)(x) - \left(\frac{1}{2}|\nabla u_{\tilde\theta}|^2 - fu_{\tilde\theta}\right)(x) \\ &= \frac{1}{2}\sum_{i=1}^d (\partial_{x_i}u_\theta(x) + \partial_{x_i}u_{\tilde\theta}(x))(\partial_{x_i}u_\theta(x) - \partial_{x_i}u_{\tilde\theta}(x)) - f(x)(u_\theta(x) - u_{\tilde\theta}(x)). \end{aligned}$$

The Lipschitz estimate of the function class $\mathcal{F}_i$ follows using (B.5), (B.6), and (B.7):

$$\begin{aligned} |h_i(u_\theta;x) - h_i(u_{\tilde\theta};x)| &\le \left(d\left(\prod_{k=1}^{L-1}n_k\right)^3 R^{3L-1+}(L+1)\sqrt{\boldsymbol{n}_L} + \sqrt{\boldsymbol{n}_L}\left(\prod_{k=1}^{L-1}n_k\right)R^{L-1}\|f\|_{L^\infty(\Omega)}\right) \times \|\theta - \tilde\theta\|_{\ell^2} \\ &\le \left(d\boldsymbol{n}_L^{3L-5/2}R^{3L-1}(L+1) + \boldsymbol{n}_L^{L-1/2}R^{L-1}\|f\|_{L^\infty(\Omega)}\right) \times \|\theta - \tilde\theta\|_{\ell^2} =: \Lambda_i\|\theta - \tilde\theta\|_{\ell^2}. \end{aligned}$$

The Lipschitz estimate of the function class $\mathcal{F}_b$ is derived using (B.4) and (B.5) as follows:

$$\begin{aligned} |h_b(u_\theta;x) - h_b(u_{\tilde\theta};x)| &= |(u_\theta - g)(x)^2 - (u_{\tilde\theta} - g)(x)^2| = |(u_\theta + u_{\tilde\theta} - 2g)(x)| \times |(u_\theta - u_{\tilde\theta})(x)| \\ &\le 2\left((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)}\right)\sqrt{\boldsymbol{n}_L}\left(\prod_{k=1}^{L-1}n_k\right)R^{L-1}\|\theta - \tilde\theta\|_{\ell^2} \\ &\le 2\left((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)}\right)\boldsymbol{n}_L^{L-\frac{1}{2}}R^{L-1}\|\theta - \tilde\theta\|_{\ell^2} =: \Lambda_b\|\theta - \tilde\theta\|_{\ell^2}. \end{aligned} \tag{3.25}$$

An application of (B.10) with $\beta = \frac{\lambda}{2}$ and a use of M_i, M_b, Λ_i, and Λ_b conclude the proof. □

Theorem 3.1 *(error estimate) Let $\varepsilon > 0$ be a given tolerance. Let ρ be sigmoid function $\frac{1}{1+e^{-x}}$ or hyperbolic tangent function $\frac{e^x - e^{-x}}{e^x + e^{-x}}$. Let $u, u_\lambda \in H^2(\Omega)$ solve* (3.1) *and* (3.2)*, respectively. Let $u_\theta^{\mathcal{S}}$ be the solution of the problem* (3.5) *generated by a random solver $\mathcal{S}$. Then for any $\mu \in (0,1)$ set the parametrized neural network function classes $\mathcal{U} = \mathcal{N}_\rho\left(c\log(d+2), c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}, c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}\right)$, the number of interior sample points $n = c(d)\varepsilon^{-Cd\log(d)}$ (resp. boundary sample points $m = c(d)\varepsilon^{-Cd\log(d)}$), $\lambda = \mathcal{O}(\varepsilon^{-1})$, and the optimization error $\mathcal{E}_{opt} \le \varepsilon^2$. Then*

$$\mathbb{E}_{\{\{X_k\}_{k=1}^n, \{Y_k\}_{k=1}^m\}}[\|u - u_\theta^*\|_{H^1(\Omega)}] \lesssim \varepsilon,$$

where the constant in $\lesssim$ depends on d, $\|f\|_{L^\infty(\Omega)}$, $\|g\|_{L^\infty(\partial\Omega)}$, and constant from the trace inequality.

Proof Recall the key estimate from Lemma 3.1 that shows

$$\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\|u-u_{\theta^*}\|_{H^1(\Omega)}] \lesssim \big(\frac{1}{\lambda^2}+\mathcal{E}_{stat}+\lambda\mathcal{E}_{app}+\mathcal{E}_{opt}\big)^{1/2}.$$

Lemma 3.2 implies that there exists neural network function class

$$\mathcal{U}=\mathcal{N}_\rho\Big(c\log(d+2), c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}, c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}\Big)$$

such that $\mathcal{E}_{app} \le c\lambda^2\varepsilon^5$. Substitute $L = c\log(d+3)$, $\boldsymbol{n}_L = c(d)\varepsilon^{-\frac{5d}{2(1-\mu)}}$, $R = c(d)\varepsilon^{-\frac{5(9d+8)}{4(1-\mu)}}$ and set $n = c(d)\varepsilon^{-Cd\log(d)}$ and $m = c(d)\varepsilon^{-Cd\log(d)}$ in Lemma 3.3 to obtain $\mathcal{E}_{stat} \lesssim \lambda\varepsilon^3$, where the constant absorbed in $\lesssim$ depends on d, $\|f\|_{L^\infty(\Omega)}$, and $\|g\|_{L^\infty(\partial\Omega)}$. A combination of all this

with $\lambda = \mathcal{O}(\varepsilon^{-1})$ and the assumption that the optimization error is controlled by ε^2 completes the proof. □

4 Physics informed neural network

For $f \in C(\bar{\Omega})$ and $g \in C(\partial\Omega)$ the continuous loss functional $\mathcal{L}(u)$ in PINN that corresponds to (1.1) reads

$$\mathcal{L}(u) = \|\Delta u + f\|^2_{L^2(\Omega)} + \gamma\|u-g\|^2_{L^2(\partial\Omega)}, \tag{4.1}$$

where $\gamma > 0$ serves as a weight for the penalty term that enforces the Dirichlet boundary condition. Note that the parameter γ is different from λ in DRM in the sense that γ is chosen as a fixed penalty parameter in this case unlike in DRM, where the convergence with respect to the parameter λ is considered.

The empirical loss function $\widehat{\mathcal{L}}(u_\theta)$ for $u_\theta \in \mathcal{U} := \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ in this case reads:

$$\widehat{\mathcal{L}}(u_\theta) = \frac{|\Omega|}{n}\sum_{i=1}^n |(\Delta u_\theta + f)(X_i)|^2 + \gamma\frac{|\partial\Omega|}{m}\sum_{j=1}^m |(u_\theta - g)(Y_j)|^2. \tag{4.2}$$

Note that in the above neural network function class, the activation function ρ is chosen to be hyperbolic tangent and sigmoid, and in the loss function, the derivatives of the neural network u_θ are computed using the chain rule of the derivative. In practical computations, we use automatic differentiation technique such as `torch.autograd` in PyTorch.

The PINN formulation of (1.1) is the optimization problem that seeks $u_{\theta^*} \in \mathcal{U}$ such that

$$u_{\theta^*} \in \operatorname*{argmin}_{u_\theta\in\mathcal{U}} \widehat{\mathcal{L}}(u_\theta). \tag{4.3}$$

The existence of a solution to (4.3) is ensured by the same argument presented in Remark 3.1.

4.1 Error analysis

To carry out the error analysis of the PINN method, we follow the ideas presented in [17, 29, 42].

Theorem 4.1 *(error estimate) Let $\varepsilon > 0$ be a given tolerance. Let u (resp. u_{θ^*}) be the solution of* (1.1) (*resp.* (4.3)) *and $u_\theta^{\mathcal{S}}$ be the approximation to* (4.3) *generated by a random solver $\mathcal{S}$. Assume $u \in H^3(\Omega)$. Then for any*

$\mu \in (0,1)$, set the parametrized neural network function classes $\mathcal{U} = \mathcal{N}_\rho\Big(c\log(d+3), c(d)\varepsilon^{-\frac{d}{1-\mu}}, c(d)\varepsilon^{-\frac{9d+12}{2-2\mu}}\Big)$, ρ as hyperbolic tangent or sigmoid function, the number of interior sample points $n = c(d)\varepsilon^{-Cd\log(d)}$, number of boundary sample points $m = c(d)\varepsilon^{-Cd\log(d)}$, and the optimization error $\mathcal{E}_{opt} \le \varepsilon^2$. Then

$$\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\|u - u_{\theta^*}\|_{L^2(\Omega)}] \lesssim \varepsilon, \tag{4.4}$$

where $\{X_k\}_{k=1}^n$ and $\{Y_k\}_{k=1}^m$ are interior and boundary collocation points respectively, and the constant in $\lesssim$ depends on γ, Ω d, $\|f\|_{L^\infty(\Omega)}$, $\|g\|_{L^\infty(\partial\Omega)}$, and $\|u\|_{H^3(\Omega)}$.

Proof A triangle inequality shows

$$\|u - u_{\theta^*}\|_{L^2(\Omega)} \le \left\|u - u_\theta^S\right\|_{L^2(\Omega)} + \left\|u_\theta^S - u_{\theta^*}\right\|_{L^2(\Omega)}. \tag{4.5}$$

We focus on the first term that leads to a combination of approximation, statistical, and optimization errors. The second term in (4.5) is a part of the optimization error. The rest of the proof is split into four steps: *Step 1* provides a key estimate that splits the error into contributions from various errors, *Step 2* (resp. *Step 3*) bounds the approximation (resp. statistical) error, and *Step 4* consolidates the error estimate.

Step 1 (control of $\left\|u - u_\theta^S\right\|_{L^2(\Omega)}^2$). The Poisson problem (1.1) satisfies [4, Theorem 4.2]:

$$\|u\|_{L^2(\Omega)} \lesssim \|f\|_{L^2(\Omega)} + \|g\|_{L^2(\partial\Omega)}, \tag{4.6}$$

where the constant absorbed in $\lesssim$ depends on Ω. The error $e_u := u - u_\theta^S$ satisfies

$$-\Delta e_u = f + \Delta u_\theta^S \quad \text{in } \Omega, \quad e_u = g - u_\theta^S \quad \text{on } \partial\Omega. \tag{4.7}$$

Apply the stability result (4.6) for the solution of the Poisson problem in (4.7) to arrive at $\|e_u\|_{L^2(\Omega)} \lesssim \left\|f + \Delta u_\theta^S\right\|_{L^2(\Omega)} + \left\|u_\theta^S - g\right\|_{L^2(\partial\Omega)}$. Algebraic manipulations introducing γ show

$$\left\|u - u_\theta^S\right\|_{L^2(\Omega)}^2 \lesssim \left\|f + \Delta u_\theta^S\right\|_{L^2(\Omega)}^2 + \gamma\left\|u_\theta^S - g\right\|_{L^2(\partial\Omega)}^2 \lesssim \mathcal{L}(u_\theta^S), \tag{4.8}$$

where the constant in $\lesssim$ depends on $\frac{1}{\gamma}$ and Ω. This and $\mathcal{L}(u) = 0$ yield

$$\left\|u - u_\theta^S\right\|_{L^2(\Omega)}^2 \lesssim \mathcal{L}(u_\theta^S) - \mathcal{L}(u). \tag{4.9}$$

Now analogous to (3.18), we split the term $\mathcal{L}(u_\theta^S) - \mathcal{L}(u)$ as a combination that would lead to approximation error, statistical error, and optimization error as:

$$\begin{aligned}\mathcal{L}(u_\theta^S) - \mathcal{L}(u) &= (\mathcal{L}(u_\theta^S) - \widehat{\mathcal{L}}(u_\theta^S)) + (\widehat{\mathcal{L}}(u_\theta^S) - \widehat{\mathcal{L}}(u_{\theta^*})) + (\widehat{\mathcal{L}}(u_{\theta^*}) - \widehat{\mathcal{L}}(u_\theta))\\ &\quad + (\widehat{\mathcal{L}}(u_\theta) - \mathcal{L}(u_\theta)) + (\mathcal{L}(u_\theta) - \mathcal{L}(u))\\ &\le \sup_{u_\theta\in\mathcal{U}}(\mathcal{L}(u_\theta) - \widehat{\mathcal{L}}(u_\theta)) + (\widehat{\mathcal{L}}(u_\theta^S) - \widehat{\mathcal{L}}(u_{\theta^*})) + \sup_{u_\theta\in\mathcal{U}}(\widehat{\mathcal{L}}(u_\theta) - \mathcal{L}(u_\theta)) + (\mathcal{L}(u_\theta) - \mathcal{L}(u)),\end{aligned} \tag{4.10}$$

for all $u_\theta \in \mathcal{U}$. The inequality in the last step follows from the minimizing property of u_{θ^*}, i.e., $\widehat{\mathcal{L}}(u_{\theta^*}) - \widehat{\mathcal{L}}(u_\theta) \le 0$. Moreover using (1.1), $\mathcal{L}(u) = 0$, and trace inequality we have

$$\mathcal{L}(u_\theta) - \mathcal{L}(u) \le \|\Delta(u_\theta - u)\|_{L^2(\Omega)}^2 + \gamma\|u_\theta - u\|_{L^2(\partial\Omega)}^2 \lesssim \|u - u_\theta\|_{H^2(\Omega)}^2. \tag{4.11}$$

The constant in $\lesssim$ depends on γ and trace inequality constant. Consequently, utilizing (4.11) in (4.10), and substituting the final result in (4.9) we obtain the following estimate:

$$\left\|u-u_\theta^S\right\|_{L^2(\Omega)}^2 \lesssim \sup_{u_\theta\in\mathcal{U}}(\mathcal{L}(u_\theta)-\widehat{\mathcal{L}}(u_\theta))+\widehat{\mathcal{L}}(u_\theta^{\mathcal{S}})-\widehat{\mathcal{L}}(u_{\theta^*})+\sup_{u_\theta\in\mathcal{U}}(\widehat{\mathcal{L}}(u_\theta)-\mathcal{L}(u_\theta)) + \inf_{u_\theta\in\mathcal{U}}\|u-u_\theta\|_{H^2(\Omega)}^2. \tag{4.12}$$

Elementary manipulations with (4.5) and (4.12), expectations taken with respect to collocation points $\{X_k\}_{k=1}^n$ and $\{Y_k\}_{k=1}^m$ lead to

$$\begin{aligned}\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\left\|u-u_\theta^S\right\|_{L^2(\Omega)}^2] \lesssim\ & \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\sup_{u_\theta\in\mathcal{U}}(\mathcal{L}(u_\theta)-\widehat{\mathcal{L}}(u_\theta))+\sup_{u_\theta\in\mathcal{U}}(\widehat{\mathcal{L}}(u_\theta)-\mathcal{L}(u_\theta))]\\ &+\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\widehat{\mathcal{L}}(u_\theta^{\mathcal{S}})-\widehat{\mathcal{L}}(u_{\theta^*})]+\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\left\|u_\theta^S-u_{\theta^*}\right\|_{L^2(\Omega)}]+\inf_{u_\theta\in\mathcal{U}}\|u-u_\theta\|_{H^2(\Omega)}^2. \end{aligned}\tag{4.13}$$

Utilize $(\mathbb{E}[f])^2 \le \mathbb{E}[f^2]$ on the left-hand side of (4.13) and take square root on both sides to observe $\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\|u-u_{\theta^*}\|_{L^2(\Omega)}] \lesssim (\mathcal{E}_{app}+\mathcal{E}_{sta}+\mathcal{E}_{opt})^{\frac{1}{2}}$, where the constant in $\lesssim$ depends on $\frac{1}{\gamma}$, Ω, and the constant from the trace inequality. The approximation error $\mathcal{E}_{app}$, statistical error $\mathcal{E}_{stat}$, and the optimization error $\mathcal{E}_{opt}$ are defined by:

$$\mathcal{E}_{app} := \inf_{u_\theta\in\mathcal{U}}\|u-u_\theta\|_{H^2(\Omega)}^2, \quad \mathcal{E}_{stat} := \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\sup_{u_\theta\in\mathcal{U}}(\mathcal{L}(u_\theta)-\widehat{\mathcal{L}}(u_\theta))+\sup_{u_\theta\in\mathcal{U}}(\widehat{\mathcal{L}}(u_\theta)-\mathcal{L}(u_\theta))], \text{ and}$$
$$\mathcal{E}_{opt} := \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\widehat{\mathcal{L}}(u_\theta^{\mathcal{S}})-\widehat{\mathcal{L}}(u_{\theta^*})+\left\|u_\theta^S-u_{\theta^*}\right\|_{L^2(\Omega)}^2].$$

Step 2 (control of $\mathcal{E}_{app} := \inf_{u_\theta\in\mathcal{U}}\|u-u_\theta\|_{H^2(\Omega)}^2$). Elementary manipulations show

$$\mathcal{E}_{app} = \|u\|_{H^3(\Omega)}^2 \inf_{u_\theta\in\mathcal{U}}\left\|\frac{u}{\|u\|_{H^3(\Omega)}}-\frac{u_\theta}{\|u\|_{H^3(\Omega)}}\right\|_{H^2(\Omega)}^2 = \|u\|_{H^3(\Omega)}^2 \inf_{u_\theta\in\mathcal{U}}\left\|\frac{u}{\|u\|_{H^3(\Omega)}}-u_\theta\right\|_{H^2(\Omega)}^2.$$

Apply the approximation result Lemma A.1 with $m=3$, $p=2$, $k=2$, and $\delta=\varepsilon$. This implies that there exists a neural network function u_θ of number of layers $L = c\log(d+3)$, number of non-zero parameters $\boldsymbol{n}_L = c(d)\varepsilon^{-\frac{d}{1-\mu}}$, and bounds on the parameters $R = c(d)\varepsilon^{-\frac{9d+12}{2-2\mu}}$, that is,

$$u_\theta \in \mathcal{N}_\rho\left(c\log(d+3), c(d)\varepsilon^{-\frac{d}{1-\mu}}, c(d)\varepsilon^{-\frac{9d+12}{2-2\mu}}\right)$$

such that $\left\|\frac{u}{\|u\|_{H^3(\Omega)}}-u_\theta\right\|_{H^2(\Omega)} \le \varepsilon$ for a fixed tolerance $\varepsilon\in(0,1)$. A combination of all this shows $\mathcal{E}_{app} = \inf_{u_\theta\in\mathcal{U}}\|u-u_\theta\|_{H^2(\Omega)}^2 \le \|u\|_{H^3(\Omega)}^2\varepsilon^2 \lesssim \varepsilon^2$ with the regularity assumption on the exact solution applied in the last step.

Step 3 (control of $\mathcal{E}_{stat} := \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\sup_{u_\theta\in\mathcal{U}}(\mathcal{L}(u_\theta)-\widehat{\mathcal{L}}(u_\theta))+\sup_{u_\theta\in\mathcal{U}}(\widehat{\mathcal{L}}(u_\theta)-\mathcal{L}(u_\theta))]$). This step deals with a bound for the statistical error in terms of the number of samples n, m, respectively drawn from Ω and $\partial\Omega$, the number of non-zero entries $\mathbf{n}_L$ in the weight matrices and bias vectors, and the bounds of the weights R. For the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$ defined by

$$\mathcal{F}_i = \{h_i(u_\theta;\cdot):\Omega\to\mathbb{R} \mid h_i(u_\theta;x) = (\Delta u_\theta+f)^2(x), u_\theta\in\mathcal{U}\} \text{ and}$$

$$\mathcal{F}_b = \{h_b(u_\theta;\cdot) : \partial\Omega \to \mathbb{R} \mid h_b(u_\theta;x) = (u_\theta - g)^2(x), u_\theta \in \mathcal{U}\},$$

apply (B.10) (with $\beta = \gamma$) and obtain the statistical error (see Appendix B). We first show that the boundedness constant in the hypotheses **(H1)** is given by

$$M_i := (dL\boldsymbol{n}_L^{2(L-1)}R^{2L} + \|f\|_{L^\infty(\Omega)})^2 \text{ and } M_b := ((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)})^2$$

and the Lipschitz continuity constants in **(H2)** are

$$is a \Lambda_i = 4dL^2\eta\sqrt{\boldsymbol{n}_L}\boldsymbol{n}_L^{3L-3}R^{3L-3}\big(dL\boldsymbol{n}_L^{2(L-1)}R^{2L} + \|f\|_{L^\infty(\Omega)}\big) \text{ and } \Lambda_b = 2((n_{L-1}+1)R + \|g\|_{L^\infty(\partial\Omega)})\boldsymbol{n}_L^{L-\frac{1}{2}}R^{L-1}.$$

The constant η in Λ_i above are 2 (resp. 1) for the activation function hyperbolic tangent (resp. sigmoid). For $h_i(u_\theta;\cdot) \in \mathcal{F}_i$, we have $h_i(u_\theta;x) = (\Delta u_\theta + f)^2(x)$. Apply (B.8) to obtain

$$|h_i(u_\theta;x)| = \Big(\sum_{i=1}^{d}\partial_{x_i^2}^2 u_\theta(x) + f(x)\Big)^2 \le (dL\boldsymbol{n}_L^{2(L-1)}R^{2L} + \|f\|_{L^\infty(\Omega)})^2 =: M_i.$$

For $h_b(u_\theta;\cdot) \in \mathcal{F}_b$, the bound M_b is straight forward from the definition $h_b(u_\theta;\cdot) := (u_\theta - g)^2$ and (B.4). Elementary manipulations using the definition of $h_i(u_\theta;x)$ and $h_i(u_{\tilde\theta};x)$ show

$$\begin{aligned}
&|h_i(u_\theta;x) - h_i(u_{\tilde\theta};x)|\\
&= |(\Delta u_\theta(x) + \Delta u_{\tilde\theta}(x) + 2f(x))(\Delta u_\theta(x) - \Delta u_{\tilde\theta}(x))|\\
&\le 2\big(dL\boldsymbol{n}_L^{2(L-1)}R^{2L} + \|f\|_{L^\infty(\Omega)}\big)|\Delta u_\theta(x) - \Delta u_{\tilde\theta}(x)|.
\end{aligned}$$

Then by (B.9), we deduce

$$|h_i(u_\theta;x) - h_i(u_{\tilde\theta};x)| \le 4dL^2\eta\sqrt{\boldsymbol{n}_L}\boldsymbol{n}_L^{3L-3}R^{3L-3}\big(dL\boldsymbol{n}_L^{2(L-1)}R^{2L} + \|f\|_{L^\infty(\Omega)}\big)\|\theta - \tilde\theta\|_{\ell^2} := \Lambda_i\|\theta - \tilde\theta\|_{\ell^2}.$$

The Lipschitz estimate of the function class $\mathcal{F}_b$ is already derived in (3.25). An application of (B.10) with $\beta = \gamma$ and a use of M_i, M_b, Λ_i, and Λ_b conclude that

$$\mathcal{E}_{stat} \lesssim n^{-\frac{1}{4}}\boldsymbol{n}_L^{\frac{23L}{4}-\frac{9}{2}}R^{\frac{23L}{4}-\frac{1}{4}} + \gamma m^{-\frac{1}{4}}\boldsymbol{n}_L^{\frac{L}{2}+\frac{15}{4}}R^{\frac{L}{2}+\frac{5}{2}}, \tag{4.14}$$

where the constant in $\lesssim$ depends on d, $\|f\|_{L^\infty(\Omega)}$, and $\|g\|_{L^\infty(\partial\Omega)}$.

Step 4 (consolidation). In this step, we consolidate all the results in *Steps* 1-3 to obtain the final estimate (4.4). Recall that *Step 1* shows $\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\|u - u_{\theta^*}\|_{L^2(\Omega)}] \lesssim (\mathcal{E}_{app} + \mathcal{E}_{stat} + \mathcal{E}_{opt})^{\frac{1}{2}}$. *Step 2* provides that there exists a neural network function class $\mathcal{U} = \mathcal{N}_\rho\Big(c\log(d+3), c(d)\varepsilon^{-\frac{d}{1-\mu}}, c(d)\varepsilon^{-\frac{9d+12}{2-2\mu}}\Big)$ such that $\mathcal{E}_{app} \lesssim \varepsilon^2$. Substitute $L = c\log(d+3)$, $\boldsymbol{n}_L = c(d)\varepsilon^{-\frac{d}{1-\mu}}$, $R = c(d)\varepsilon^{-\frac{9d+12}{2-2\mu}}$ in (4.14) and set $n = c(d)\varepsilon^{-Cd\log(d)}$, $m = c(d)\varepsilon^{-Cd\log(d)}$ to obtain $\mathcal{E}_{stat} \lesssim \varepsilon^2$, where the constant absorbed in $\lesssim$ depends on d, $\|f\|_{L^\infty(\Omega)}$, and $\|g\|_{L^\infty(\partial\Omega)}$. A combination of all this and the assumption that $\mathcal{E}_{opt}$ is controlled by ε^2 complete the proof. □

5 Variational PINN

The Petrov-Galerkin formulation that corresponds to (1.1) serves as the foundation for the VPINN. In VPINN, the test functions still belong to polynomial spaces but the solution is represented by a nonlinear approximation

via a neural network. Unlike the PINN that incorporates the strong form of the equation into the loss function, and DRM, which minimizes the energy functional, VPINN incorporates the weak formulation of the problem and constructs a variational loss function. This method was first introduced in [30], with subsequent developments presented in [6], among others.

Let Ω be a domain in $\mathbb{R}^d$ $(d = 2, 3)$, $f \in C(\bar{\Omega})$ and $g \in H^{1/2}(\partial\Omega)$. Let $u_g \in H^1(\Omega)$ be an extension of g in Ω. Then the weak formulation of (1.1) seeks $u \in u_g + H_0^1(\Omega)$ such that

$$a(u, v) = F(v) \quad \text{for all } v \in H_0^1(\Omega), \tag{5.1}$$

where the bilinear form $a : H^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ is defined as $a(v, w) := \int_\Omega \nabla v \cdot \nabla w \, dx$ and the linear form $F : H_0^1(\Omega) \to \mathbb{R}$ is defined as $F(w) := \int_\Omega f w \, dx$ for all $v \in H^1(\Omega)$, $w \in H_0^1(\Omega)$. We assume that we can represent u in the form

$$u = u_g + \Phi\tilde{u}, \tag{5.2}$$

for some known smooth function $\Phi \in W_0^{1,\infty}(\Omega)$ and some $\tilde{u} \in H^1(\Omega)$ that has the same smoothness as u. Introduce the affine mapping

$$B : H^1(\Omega) \to u_g + H_0^1(\Omega) \quad \text{such that} \quad Bw = u_g + \Phi w \tag{5.3}$$

which enforces the given Dirichlet boundary condition. Then (5.1) can be equivalently formulated as follows: Find $\tilde{u} \in H^1(\Omega)$ such that

$$a(B\tilde{u}, v) = F(v) \quad \text{for all } v \in H_0^1(\Omega) \tag{5.4}$$

with $u = B\tilde{u} = u_g + \Phi\tilde{u}$. In VPINN, we use neural networks to approximate the trial function and finite element functions for the test functions. Let $\mathcal{T}_h$ be a quasi-uniform shape-regular mesh of the domain Ω. Let $h_T = \text{diam}(T)$ for $T \in \mathcal{T}_h$ and let the mesh size h be defined as $h = \max\{h_T \mid T \in \mathcal{T}_h\}$. Let $V_h := \{v_h \in H^1(\Omega) \mid v_h|_T \in \mathbb{P}_k(T) \quad \text{for all } T \in \mathcal{T}_h\}$ be the space of trial functions and $V_{h,0} = V_h \cap H_0^1(\Omega)$ be the approximate test function space. Furthermore, introduce computable approximations of the forms a and F by numerical quadratures. Precisely, for any $T \in \mathcal{T}_h$, let $\left\{(\xi_l^T, \omega_l^T) : l \in I^T\right\}$ be the nodes and weights of a quadrature formula of precision $q \geq 2k$ on T. We define the approximate forms

$$a_h(w, v) = \sum_{T\in\mathcal{T}_h} \sum_{l\in I^T} \omega_l^T (\nabla w(\xi_l^T) \cdot \nabla v(\xi_l^T)) \quad \text{and} \quad F_h(v) = \sum_{T\in\mathcal{T}_h} \sum_{l\in I^T} \omega_l^T f(\xi_l^T) v(\xi_l^T). \tag{5.5}$$

With these ingredients at hand, we would like to approximate the solution of (5.4) by some $u_\theta \in \mathcal{U} := \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ satisfying

$$a_h(\mathcal{I}_h B u_\theta, v_h) = F_h(v_h) \quad \text{for all } v_h \in V_{h,0}. \tag{5.6}$$

Note that the choices of activation functions ρ for this method are hyperbolic tangent and sigmoid. Here $\mathcal{I}_h : \mathrm{C}^0(\bar{\Omega}) \to V_h$ (with $\mathcal{I}_h : \mathrm{C}^0(\bar{\Omega}) \cap H_0^1(\Omega) \to V_{h,0}$) is an interpolation operator such that it satisfies the interpolation estimate

$$|v - \mathcal{I}_h v|_{\ell,T} \lesssim h^{k+1-\ell} |v|_{k+1,T}, \quad 0 \leq \ell \leq k+1 \quad \text{for all } T \in \mathcal{T}_h \tag{5.7}$$

(see for example, [21, subsection 11.5.1]). In order to propose VPINN, introduce a basis $\{\varphi_i : i \in I_h\}$ of $V_{h,0}$, and for any w smooth enough, define the residual as follows:

$$r_{h,i}(w) = F_h(\varphi_i) - a_h(\mathcal{I}_h B w, \varphi_i), \quad i \in I_h, \tag{5.8}$$

where I_h is the global basis index set. The VPINN formulation of (1.1) is the optimization problem that seeks $u_{\theta^*} \in \mathcal{U} = \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$ such that

$$u_{\theta^*} \in \arg\min_{u_\theta \in \mathcal{U}} \widehat{\mathcal{L}}_h(u_\theta) := \sum_{i \in I_h} r_{h,i}^2(u_\theta). \tag{5.9}$$

Note that the VPINN needs basis functions of the finite element space V_h. Meshing and computing basis functions in higher dimensions ($\mathbb{R}^d,\ d \geq 4$) is a challenge and visualisation is impossible. Therefore, for VPINN, we restrict ourselves to two or three-dimensional domains.

5.1 Error analysis

To carry out the error analysis of the VPINN, we follow the ideas presented in [6]. Let $u_\theta^{\mathcal{S}} \in \mathcal{U}$ be a solution of the minimization problem (5.9) using a random solver $\mathcal{S}$ and set

$$u_\theta^{\mathcal{S},h} = \mathcal{I}_h B u_\theta^{S} \in V_h. \tag{5.10}$$

Recall the definition (5.3) of the affine mapping B to observe

$$u_\theta^{\mathcal{S},h} = \mathcal{I}_h u_g + \mathcal{I}_h(\Phi u_\theta^{\mathcal{S}}), \tag{5.11}$$

where $\mathcal{I}_h u_g$ is a discrete lifting in V_h of the Dirichlet data g and $\mathcal{I}_h(\Phi u_\theta^{\mathcal{S}}) \in V_{h,0}$. We aim to estimate the error between u and u_{θ^*} in the energy norm. To accomplish this task, we need several results. We start with the assumptions **(A1)-(A4)** stated below.

- **(A1)** (quadrature) The quadrature rules used in the elements in $\mathcal{T}_h$ are obtained by affine transformations from a quadrature rule $\{(\hat{\xi}_\iota, \hat{\omega}_\iota) : \iota \in \hat{I}\}$ on a reference element $\hat{E} \subset \mathbb{R}^n$.
- **(A2)** (mesh regularity) The mesh $\mathcal{T}_h$ is quasi uniform.
- **(A3)** (data smoothness) The source term f belongs to $W^{k,\infty}(\Omega)$ and the boundary data g belongs to $H^{k+\frac{1}{2}}(\partial\Omega)$ with $k \geq 2$.
- **(A4)** (solution smoothness) Let $u_g \in H^{k+1}(\Omega)$, $\tilde{u} \in H^{k+1}(\Omega)$ and $\Phi \in \mathrm{W}_0^{k+1,\infty}(\Omega)$. Furthermore, the manifold formed by the neural network functions satisfies the smoothness condition $\mathcal{U} \subset H^2(\Omega)$.

Clearly, from **(A4)**, we have $u = u_g + \Phi\tilde{u} \in H^{k+1}(\Omega)$. The next proposition on norm equivalence follows from [13, Theorem 4.1.4].

Lemma 5.1 (norm equivalence) *[6, Definition 1, page 100] Let $v_h \in V_h$ such that $v_h = \sum_{i=1}^{|I_h|} v_i \phi_i$, where $\{\phi_i\}_{i=1}^{|I_h|}$ is a Lagrange basis of the space V_h. Then it holds that*

$$h^{1-\frac{d}{2}} \|\nabla v_h\|_{L^2(\Omega)} \lesssim \|\boldsymbol{v}\|_{\ell^2} \lesssim h^{-\frac{d}{2}} \|v_h\|_{H^1(\Omega)}, \tag{5.12}$$

where $\boldsymbol{v} = (v_i)_{i \in I_h}$, and $\|\boldsymbol{v}\|_{\ell^2} = (\sum_{i \in I_h} v_i^2)^{1/2}$.

For all $v_h, w_h \in V_h$, we introduce the consistency errors due to numerical quadratures as:

$$E_h^a(w_h, v_h) = a(w_h, v_h) - a_h(w_h, v_h) \quad \text{and} \quad E_h^f(v_h) = F(v_h) - F_h(v_h). \tag{5.13}$$

Proposition 5.1 *(approximation of the forms a and f) For all $w_h, v_h \in V_h$, under the assumption* **(A3)**, *the numerical quadratures satisfy the bounds*

$$|E_h^a(w_h, v_h)| \lesssim h^k \|w_h\|_{H^k(\mathcal{T}_h)} \|v_h\|_{H^1(\Omega)} \quad \text{and} \quad |E_h^f(v_h)| \lesssim h^k \, \|f\|_{W^{k,\infty}(\Omega)} \, \|v_h\|_{H^1(\Omega)}, \tag{5.14}$$

where the constant in $\lesssim$ is independent of the mesh size h.

Proposition 5.2 *(discrete inf-sup condition) Under the assumption* **(A3)**, *for all $h \le h_0$ small enough, the bilinear form a_h satisfies an inf-sup condition with respect to the space V_h. That is, there exists a constant $\tilde{\alpha}_\star > 0$ such that*

$$\tilde{\alpha}_\star \|w_h\|_{H^1(\Omega)} \le \sup_{v_h \in V_{h,0}} \frac{a_h(w_h, v_h)}{\|v_h\|_{H^1(\Omega)}} \quad \text{for all } w_h \in V_{h,0}. \tag{5.15}$$

Proof Since (5.1) is well-posed and $V_h \subset H^1(\Omega)$ (resp. $V_{h,0} \subset H_0^1(\Omega)$), the coercivity of the bilinear form a on $V_{h,0}$ shows the existence of $\alpha_* > 0$ such that

$$\alpha_\star \|w_h\|_{H^1(\Omega)} \le \sup_{v_h \in V_{h,0}} \frac{a(w_h, v_h)}{\|v_h\|_{H^1(\Omega)}} \quad \text{for all } w_h \in V_{h,0}. \tag{5.16}$$

Moreover, we have $a_h(w_h, v_h) = a(w_h, v_h) - E_h^a(w_h, v_h)$ for all $w_h, v_h \in V_h$. This and the first identity in (5.14) with $k = 2$ yields

$$\alpha_\star \|w_h\|_{H^1(\Omega)} \le \sup_{v_h \in V_{h,0}} \frac{a_h(w_h, v_h)}{\|v_h\|_{H^1(\Omega)}} + ch^2 \, \|w_h\|_{H^1(\Omega)} \quad \text{for all } w_h \in V_{h,0}, \tag{5.17}$$

where $c > 0$ is a constant independent of the mesh size h. Choose $h_0 > 0$ small enough such that $ch^2 \le \frac{1}{2}\alpha_\star$ for all $h \le h_0$. This concludes the proof with $\tilde{\alpha}_\star = \frac{1}{2}\alpha_\star$. □

Theorem 5.1 *(key estimate) Let $u := B\tilde{u} = u_g + \Phi\tilde{u}$ solve* (5.4) *and u_{θ^*} solve* (5.9), *that is, $Bu_{\theta^*} = u_g + \Phi u_{\theta^*}$ be an approximation of the solution u. Let $u_\theta^{\mathcal{S}}$ be a minimiser of* (5.9) *obtained by a random solver $\mathcal{S}$. Then the following error estimate holds:*

$$\|u - Bu_{\theta^*}\|_{H^1(\Omega)} \lesssim \mathcal{E}_{int} + \tilde{\alpha}_\star^{-1}(\mathcal{E}_{app} + \mathcal{E}_{quad}) + \mathcal{E}_{opt},$$

where the interpolation error $\mathcal{E}_{int} := \left(1 + \frac{1}{\tilde{\alpha}_\star}\left(1 + h^{-1}\right)\right) \|u - \mathcal{I}_h u\|_{H^1(\Omega)}$, the approximation error $\mathcal{E}_{app} := h^{-1}\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}$, the quadrature error

$$\mathcal{E}_{quad} := \sup_{v_h \in V_{h,0}} \frac{E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)}{\|v_h\|_{H^1(\Omega)}} + h^{-1} \sup_{z_h \in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}},$$

the optimization error $\mathcal{E}_{opt} := \|u_\theta^{\mathcal{S},h} - Bu_{\theta^*}\|_{H^1(\Omega)}$, and the constant in $\lesssim$ is independent of the mesh size h.

Remark 5.1 Note that $\mathcal{E}_{int}$ represents the interpolation error, $\mathcal{E}_{app}$ represents an approximation error, which measures how well the computed solution approximates the true solution, and the standard statistical error is replaced by quadrature error $\mathcal{E}_{quad}$ since we have used quadrature points to approximate the bilinear and linear forms. The optimization error measures the gap between the analytical minimiser Bu_{θ^*} and the interpolation of the computed solution $u_\theta^{\mathcal{S},h} := \mathcal{I}_h u_\theta^{\mathcal{S}}$.

Proof Introduce $u_\theta^{\mathcal{S},h}$ and $\mathcal{I}_h u$ to decompose the error using the triangle inequality as

$$\|u - Bu_{\theta^*}\|_{H^1(\Omega)} \leq \|u - \mathcal{I}_h u\|_{H^1(\Omega)} + \|\mathcal{I}_h u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} + \|u_\theta^{\mathcal{S},h} - Bu_{\theta^*}\|_{H^1(\Omega)}. \tag{5.18}$$

The first term in (5.18) is the interpolation error for which the error bounds are known, the third term is the optimization error; we estimate only the second term. The following results in *Step 1* and *Step 2* are key to derive the estimate of the second term in *Step 3*.

Step 1 (control of $\|u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)}$). Set $e^\theta := u - Bu_\theta^{\mathcal{S}}$. From (5.2) and (5.3), we have $e^\theta = \Phi(\tilde{u} - u_\theta^{\mathcal{S}})$. Thus, $u - u_\theta^{\mathcal{S},h} = (u - \mathcal{I}_h u) - e^\theta + (I - \mathcal{I}_h)e^\theta$. Apply a standard interpolation estimate for the third term, **(A4)** to see $e^\theta \in H^2(\Omega)$, $\Phi \in W^{k+1,\infty}(\Omega)$, and obtain

$$\|u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} \lesssim \|u - \mathcal{I}_h u\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + h\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}, \tag{5.19}$$

where the constant in $\lesssim$ is independent of the mesh size h.

Step 2 (control of $\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})}$). Since $\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})}$ is a weighted ℓ_2-norm in $\mathbb{R}^{|I_h|}$, we can write

$$\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} = \sup_{\mathbf{z}\in\mathbb{R}^{|I_h|}} \frac{1}{\|\mathbf{z}\|_{\ell^2}} \sum_{i\in I_h} r_{h,i}(u_\theta^{\mathcal{S}}) z_i, \tag{5.20}$$

where for $z_h = \sum_{i\in I_h} z_i\varphi_i \in V_{h,0}$, $\sum_{i\in I_h} r_{h,i}(u_\theta^{\mathcal{S}})z_i = F_h(z_h) - a_h(\mathcal{I}_h Bu_\theta^{\mathcal{S}}, z_h)$. In analogy with (5.10), let us set $u_\theta^{\mathcal{S},h} = \mathcal{I}_h Bu_\theta^{\mathcal{S}} \in V_h$. Recall (5.12) to obtain

$$\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} \leq h^{\frac{d}{2}-1} \sup_{z_h\in V_{h,0}} \frac{F_h(z_h) - a_h(u_\theta^{\mathcal{S},h}, z_h)}{\|z_h\|_{H^1(\Omega)}}. \tag{5.21}$$

Algebraic manipulations of the numerator of (5.21) yields

$$F_h(z_h) = f(z_h) - E_h^f(z_h) = a(u, z_h) - E_h^f(z_h) \text{ and } a_h(u_\theta^{\mathcal{S},h}, z_h) = a(u_\theta^{\mathcal{S},h}, z_h) - E_h^a(u_\theta^{\mathcal{S},h}, z_h). \tag{5.22}$$

Utilize (5.22) in (5.21) to arrive at

$$\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} \leq h^{\frac{d}{2}-1} \left(\|u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} + \sup_{z_h\in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}} \right). \tag{5.23}$$

Incorporate (5.19) in (5.23) to obtain

$$\sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} \lesssim h^{\frac{d}{2}-1} \left(\|u - \mathcal{I}_h u\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + h\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)} + \sup_{z_h\in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}} \right). \tag{5.24}$$

Step 3 (error estimate). Utilize (5.2) and (5.11) to write the second term of (5.18) as $\mathcal{I}_h u - u_\theta^{\mathcal{S},h} = \mathcal{I}_h(\Phi\tilde{u}) - \mathcal{I}_h(\Phi u_\theta^{\mathcal{S}}) \in V_{h,0}$. Hence, we can apply (5.15) to obtain

$$\|\mathcal{I}_h u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} \leq \frac{1}{\tilde{\alpha}_\star} \sup_{v_h \in V_{h,0}} \frac{a_h(\mathcal{I}_h u - u_\theta^{\mathcal{S},h}, v_h)}{\|v_h\|_{H^1(\Omega)}}. \tag{5.25}$$

Recall (5.13) and (5.1) to obtain

$$\begin{aligned} a_h(\mathcal{I}_h u, v_h) &= a(\mathcal{I}_h u, v_h) - E_h^a(\mathcal{I}_h u, v_h) = a(u, v_h) - a(u - \mathcal{I}_h u, v_h) - E_h^a(\mathcal{I}_h u, v_h) \\ &= F(v_h) - a(u - \mathcal{I}_h u, v_h) - E_h^a(\mathcal{I}_h u, v_h) = F_h(v_h) + E_h^f(v_h) - a(u - \mathcal{I}_h u, v_h) - E_h^a(\mathcal{I}_h u, v_h). \end{aligned} \tag{5.26}$$

This shows that the numerator in (5.25) is

$$a_h(\mathcal{I}_h u - u_\theta^{\mathcal{S},h}, v_h) = \left(F_h(v_h) - a_h(u_\theta^{\mathcal{S},h}, v_h)\right) - a(u - \mathcal{I}_h u, v_h) + \left(E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)\right). \tag{5.27}$$

The definitions of $u_\theta^{\mathcal{S},h}$ from (5.10) and the residual from (5.8) lead to

$$F_h(v_h) - a_h(u_\theta^{\mathcal{S},h}, v_h) = F_h(v_h) - a_h(\mathcal{I}_h B u_\theta^{\mathcal{S}}, v_h) = \sum_{i \in I_h} r_{h,i}(u_\theta^{\mathcal{S}}) v_i. \tag{5.28}$$

Hence, (5.20) and (5.12) yield

$$F_h(v_h) - a_h(u_\theta^{\mathcal{S},h}, v_h) \lesssim h^{-\frac{d}{2}} \sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} \|v_h\|_{H^1(\Omega)}. \tag{5.29}$$

A combination of (5.27), (5.29), and (5.25) with an application of the continuity of the bilinear form a yields

$$\|\mathcal{I}_h u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} \lesssim \tilde{\alpha}_\star^{-1} \left(\|u - \mathcal{I}_h u\|_{H^1(\Omega)} + h^{-\frac{d}{2}} \sqrt{\widehat{\mathcal{L}}_h(u_\theta^{\mathcal{S}})} + \sup_{v_h \in V_{h,0}} \frac{E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)}{\|v_h\|_{H^1(\Omega)}} \right). \tag{5.30}$$

A substitution of (5.24) in (5.30) shows

$$\begin{aligned} \|\mathcal{I}_h u - u_\theta^{\mathcal{S},h}\|_{H^1(\Omega)} \lesssim \tilde{\alpha}_\star^{-1} \Bigg(& (1 + h^{-1}) \|u - \mathcal{I}_h u\|_{H^1(\Omega)} + h^{-1} \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)} \\ & + \sup_{v_h \in V_{h,0}} \frac{E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)}{\|v_h\|_{H^1(\Omega)}} + h^{-1} \sup_{z_h \in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}} \Bigg). \end{aligned} \tag{5.31}$$

A consolidation of (5.18) and (5.31) concludes the proof. □

Theorem 5.2 *(error estimate) Let $u := B\tilde{u} = u_g + \Phi\tilde{u}$ solve* (5.4) *and u_{θ^*} solve* (5.9)*, that is, let $Bu_{\theta^*} = u_g + \Phi u_{\theta^*}$ be an approximation of the solution u. Let $u_\theta^{\mathcal{S}}$ be a minimiser of* (5.9) *obtained by a random solver $\mathcal{S}$ and let* **(A1)-(A4)** *hold. (a) Then the following error estimate holds:*

$$\begin{aligned} \|u - Bu_{\theta^*}\|_{H^1(\Omega)} \lesssim\ & \left(1 + \tilde{\alpha}_\star^{-1}\right) h^k + \tilde{\alpha}_\star^{-1} h^{k-1} + \tilde{\alpha}_\star^{-1} \left(h^{-1} \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}\right) \\ & + \tilde{\alpha}_\star^{-1} \left(\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + h\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}\right) + \tilde{\alpha}_\star^{-1} h^k \left(\|u_g\|_{H^k(\Omega)} + \|\tilde{u}\|_{H^k(\Omega)}\right) \\ & + \|u_\theta^{\mathcal{S},h} - Bu_{\theta^*}\|_{H^1(\Omega)}. \end{aligned} \tag{5.32}$$

Remark 5.2 For details of control of the terms $\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}$ and $\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)}$, we refer to [6].

Proof The key estimate in Theorem 5.1 yields

$$\|u - Bu_{\theta^*}\|_{H^1(\Omega)} \lesssim \mathcal{E}_{int} + \tilde{\alpha}_\star^{-1}(\mathcal{E}_{app} + \mathcal{E}_{quad}) + \mathcal{E}_{opt}. \tag{5.33}$$

We estimate the components $\mathcal{E}_{int}$ and $\mathcal{E}_{quad}$ in the right-hand side of (5.33) in the following two steps, and leave the approximation error $\mathcal{E}_{app} := h^{-1}\|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^1(\Omega)} + \|\tilde{u} - u_\theta^{\mathcal{S}}\|_{H^2(\Omega)}$ and the optimization error $\mathcal{E}_{opt} := \|u_\theta^{\mathcal{S},h} - Bu_{\theta^*}\|_{H^1(\Omega)}$ as it is.

Step 1 (estimate of $\mathcal{E}_{int}$). From **(A2)**, we have $u \in H^{k+1}(\Omega)$. The interpolation estimate of $\mathcal{I}_h$ yields

$$\begin{aligned}\mathcal{E}_{int} = \left(1 + \tilde{\alpha}_\star^{-1}(1 + h^{-1})\right)\|u - \mathcal{I}_h u\|_{H^1(\Omega)} &\le \left(1 + \tilde{\alpha}_\star^{-1}(1 + h^{-1})\right) h^k \|u\|_{H^{k+1}(\Omega)} \\ &= \left((1 + \tilde{\alpha}_\star^{-1})h^k + \tilde{\alpha}_\star^{-1}h^{k-1}\right)\|u\|_{H^{k+1}(\Omega)}.\end{aligned} \tag{5.34}$$

Step 2 (estimate of $\mathcal{E}_{quad}$). The quadrature error is

$$\mathcal{E}_{quad} = \sup_{v_h \in V_{h,0}} \frac{E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)}{\|v_h\|_{H^1(\Omega)}} + h^{-1} \sup_{z_h \in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}}. \tag{5.35}$$

The first and second components of $\mathcal{E}_{quad}$ are estimated using (5.14) as

$$\sup_{v_h \in V_{h,0}} \frac{E_h^f(v_h) - E_h^a(\mathcal{I}_h u, v_h)}{\|v_h\|_{H^1(\Omega)}} \le h^k(\|f\|_{W^{k,\infty}(\Omega)} + \|\mathcal{I}_h u\|_{H^k(\mathcal{T}_h)}) \tag{5.36}$$

$$h^{-1} \sup_{z_h \in V_{h,0}} \frac{E_h^a(u_\theta^{\mathcal{S},h}, z_h) - E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}} \le h^{k-1}(\|\mathcal{I}_h Bu_\theta^{\mathcal{S}}\|_{H^k(\mathcal{T}_h)} + \|f\|_{W^{k,\infty}(\Omega)}). \tag{5.37}$$

A triangle inequality, (5.3), and inverse inequality lead to

$$\begin{aligned}\|\mathcal{I}_h Bu_\theta^{\mathcal{S}}\|_{H^k(\mathcal{T}_h)} &\le \|\mathcal{I}_h B(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^k(\mathcal{T}_h)} + \|\mathcal{I}_h B\tilde{u}\|_{H^k(\mathcal{T}_h)} \\ &= \|\mathcal{I}_h \Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^k(\mathcal{T}_h)} + \|\mathcal{I}_h B\tilde{u}\|_{H^k(\mathcal{T}_h)} \\ &\le h^{1-k}\|\mathcal{I}_h \Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^1(\Omega)} + \|\mathcal{I}_h B\tilde{u}\|_{H^k(\mathcal{T}_h)}.\end{aligned} \tag{5.38}$$

A triangle inequality applied once again shows

$$\|\mathcal{I}_h \Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^1(\Omega)} \le \|\mathcal{I}_h \Phi(u_\theta^{\mathcal{S}} - \tilde{u}) - \Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^1(\Omega)} + \|\Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^1(\Omega)}.$$

This with a standard interpolation estimate (5.7) and (**A4**) for the first term and (**A4**) in the second term yields

$$\|\mathcal{I}_h \Phi(u_\theta^{\mathcal{S}} - \tilde{u})\|_{H^1(\Omega)} \lesssim h\|u_\theta^{\mathcal{S}} - \tilde{u}\|_{H^2(\Omega)} + \|u_\theta^{\mathcal{S}} - \tilde{u}\|_{H^1(\Omega)}, \tag{5.39}$$

where the absorbed constant in $\lesssim$ depends on $\|\Phi\|_{W^{k+1,\infty}(\Omega)}$ and interpolation constant. Substitute (5.39) in (5.38) to obtain

$$\|\mathcal{I}_h Bu_\theta^{\mathcal{S}}\|_{H^k(\mathcal{T}_h)} \lesssim h^{1-k}(h\|u_\theta^{\mathcal{S}} - \tilde{u}\|_{H^2(\Omega)} + \|u_\theta^{\mathcal{S}} - \tilde{u}\|_{H^1(\Omega)}) + \|\mathcal{I}_h B\tilde{u}\|_{H^k(\mathcal{T}_h)}. \tag{5.40}$$

A combination of (5.38)-(5.40) in (5.37) leads to

$$h^{-1}\sup_{z_h\in V_{h,0}}\frac{E_h^a(u_\theta^{\mathcal{S},h},z_h)-E_h^f(z_h)}{\|z_h\|_{H^1(\Omega)}}\leq \|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^1(\Omega)}+h\|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^2(\Omega)}+h^k(\|\mathcal{I}_hB\tilde{u}\|_{H^k(\mathcal{T}_h)}+\|f\|_{W^{k,\infty}(\Omega)}). \tag{5.41}$$

Utilize (5.36) and (5.41) in (5.35), and apply the interpolation estimate from (5.7) to obtain

$$\begin{aligned}\mathcal{E}_{quad}&\leq \|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^1(\Omega)}+h\|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^2(\Omega)}+h^k(\|\mathcal{I}_hB\tilde{u}\|_{H^k(\mathcal{T}_h)}+\|\mathcal{I}_hu\|_{H^k(\mathcal{T}_h)}+\|f\|_{W^{k,\infty}(\Omega)})\\&\lesssim \|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^1(\Omega)}+h\|u_\theta^{\mathcal{S}}-\tilde{u}\|_{H^2(\Omega)}+h^k(\|\tilde{u}\|_{H^k(\mathcal{T}_h)}+\|u_g\|_{H^k(\mathcal{T}_h)}+\|f\|_{W^{k,\infty}(\Omega)})\end{aligned} \tag{5.42}$$

with the stability of $\mathcal{I}_h$, the definition $u:=B\tilde{u}=u_g+\Phi\tilde{u}$, and **(A4)** in the last step. Finally substituting (5.34) and (5.42) in (5.33), we conclude (5.32). □

Remark 5.3 (**Optimisation error**) The optimisation error $\mathcal{E}_{opt}$ refers to the error introduced by the algorithm used to train neural networks for PDE problems. For DRM, the optimisation error measures the empirical loss function gap and the H^1-gap between the analytical minimiser u_{θ^*} and the minimiser $u_\theta^{\mathcal{S}}$ obtained from the random solver. It is defined as $\mathcal{E}_{opt}=\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}\big[\widehat{\mathcal{L}}(u_\theta^{\mathcal{S}})-\widehat{\mathcal{L}}(u_{\theta^*})+\|u_\theta^S-u_{\theta^*}\|^2_{H^1(\Omega)}\big]$, see Lemma 3.1. For PINN, the optimisation error measures the empirical loss function gap and the L^2-gap between the analytical minimiser u_{θ^*} and the minimiser $u_\theta^{\mathcal{S}}$ obtained from the random solver. It is defined as $\mathcal{E}_{opt}=\mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}\big[\widehat{\mathcal{L}}(u_\theta^{\mathcal{S}})-\widehat{\mathcal{L}}(u_{\theta^*})+\|u_\theta^S-u_{\theta^*}\|^2_{L^2(\Omega)}\big]$, see Step 1 of the proof of Theorem 4.1. For VPINN, the optimisation error measures the gap between the interpolation of the minimiser $u_\theta^{\mathcal{S},h}=\mathcal{I}_hBu_\theta^{\mathcal{S}}$ obtained from the random solver and the analytical minimiser Bu_{θ^*}. It is defined as $\mathcal{E}_{opt}:=\|u_\theta^{\mathcal{S},h}-Bu_{\theta^*}\|_{H^1(\Omega)}$, see Theorem 5.1. Training such networks remains highly challenging because the underlying optimisation landscape is non-convex. The theoretical understanding of optimisation error is still quite limited, except in a few simplified settings where the training dynamics become essentially linear (see, for example, [24, 36]). For PDE-based learning, one can expect difficulties of a similar or even greater magnitude.

6 Numerical implementation

This section discusses the implementation procedure for DRM/PINN/VPINN in Algorithm 1, and this is followed by numerical experiments. The codes and datasets generated and analyzed in this article are openly available in the GitHub repository `DNN_review_Article` at

https://github.com/rcs1994/DNN_review_Article

Example 6.1 (**DRM, PINN, and VPINN on the unit square domain in 2D**)

Consider the Poisson equation (1.1) in $\Omega=(0,1)^2$ with the known exact solution $u=e^{x_1(1-x_1)}\sin(\pi x_2)+e^{x_2(1-x_2)}\sin(\pi x_1)$. The source f is computed as $f=-\Delta u$ and the boundary data is chosen as $g=u$ on $\partial\Omega$.
For DRM and PINN, we use a DNN architecture with 4 hidden layers and 30 neurons in each layer. When formulating the empirical loss $\widehat{\mathcal{L}}(u_\theta)$, we choose $n=5000$ i.i.d.(independent and identically distributed) points from $U(\Omega)$ for the PDE residual, $m=1000$ i.i.d. points from $U(\partial\Omega)$ for boundary residual, and boundary weight $\alpha=700$. These settings are chosen by trial and error. We minimize the loss $\widehat{\mathcal{L}}(u_\theta)$ using ADAM provided by the PyTorch library. We use 10000 ADAM iterations to minimize the loss function and the learning rate in the stochastic gradient method is 10^{-4}.
For VPINN, we choose a triangulation with 2113 nodes and 4096 elements. The test functions are piecewise linear functions. We use another pre-trained neural network to enforce the boundary condition, having 2 hidden layers

with a width 30. It is trained on the known boundary data. For both the methods, ADAM is employed to optimize the bias and weights in neural networks. The learning rate for the first 5000 epochs is 10^{-3} and 10^{-4} for the rest.

Algorithm 1 Training Framework for DRM, PINN, and VPINN

procedure TRAIN DRM/PINN/VPINN
  Construct neural network:
    Architecture: L layers with activation ρ
    Represented function: u_θ
  Input:
  if Method is DRM or PINN **then**
    Collocation points: $\{X_i\}_{i=1}^n \subset \Omega$, $\{Y_j\}_{j=1}^m \subset \partial\Omega$
    Data: $\{f(X_i)\}_{i=1}^n$, $\{g(Y_j)\}_{j=1}^m$
    Balancing parameter: $\lambda > 0$ ($\gamma > 0$)
  else ▷ VPINN
    Mesh: $\mathcal{T}_h$
    Test space: V_h with $\mathbb{P}_k$ basis functions $\{\varphi_i\}_{i=1}^{n_b}$
  end if
  Initialize parameters: $\theta^0 \leftarrow \{(W^\ell, b^\ell)\}_{\ell=1}^L$ ▷ e.g., Xavier initialization [25]
  $k \leftarrow 0$
  while $k \leq N$ **do** ▷ Total training epochs: N
    Compute empirical loss $\widehat{\mathcal{L}}(\theta^k)$:
    if DRM **then**
      $\widehat{\mathcal{L}}_\lambda(\theta^k) \leftarrow \frac{|\Omega|}{n}\sum_{i=1}^n \left[\frac{1}{2}|\nabla u_{\theta^k}|^2 - f u_{\theta^k}\right](X_i) + \frac{\lambda}{2}\frac{|\partial\Omega|}{m}\sum_{i=1}^m |u_{\theta^k}(Y_i) - g(Y_i)|^2$
    else if PINN **then**
      $\widehat{\mathcal{L}}(\theta^k) \leftarrow \frac{|\Omega|}{n}\sum_{i=1}^n |\Delta u_{\theta^k}(X_i) + f(X_i)|^2 + \gamma\frac{|\partial\Omega|}{m}\sum_{j=1}^m |u_{\theta^k}(Y_j) - g(Y_j)|^2$
    else ▷ VPINN
      $\widehat{\mathcal{L}}_h(\theta^k) \leftarrow \sum_{i\in I_h}\left[F_h(\varphi_i) - a_h(\mathcal{I}_h(Bu_{\theta^k}), \varphi_i)\right]^2$
    end if
    Compute gradient: $g^k \leftarrow \nabla_\theta \widehat{\mathcal{L}}(\theta^k)$
    Update parameters: $\theta^{k+1} \leftarrow \text{Optimizer}(\theta^k, g^k, \tau^k)$ ▷ Optimiser: Adam [31], AdaGrad [19]
    (τ^k be the learning rate)
    $k \leftarrow k + 1$
  end while
  Return: Trained parameters θ^N and solution u_{θ^N}
end procedure

The exact and approximate solutions of all three methods are displayed in Figure 2. The L^2 and H^1-errors and relative errors are shown in Table 2. The L^2 and H^1 relative errors are defined, respectively, as $L^2-\texttt{relative error} := \|u - u_\theta\|_{L^2(\Omega)} / \|u\|_{L^2(\Omega)}$ and $H^1-\texttt{relative error} := \|u - u_\theta\|_{H^1(\Omega)} / \|u\|_{H^1(\Omega)}$. For this example, the training time (in seconds) for DRM, PINN, and VPINN are 80.03, 18.18, and 277.08, respectively.

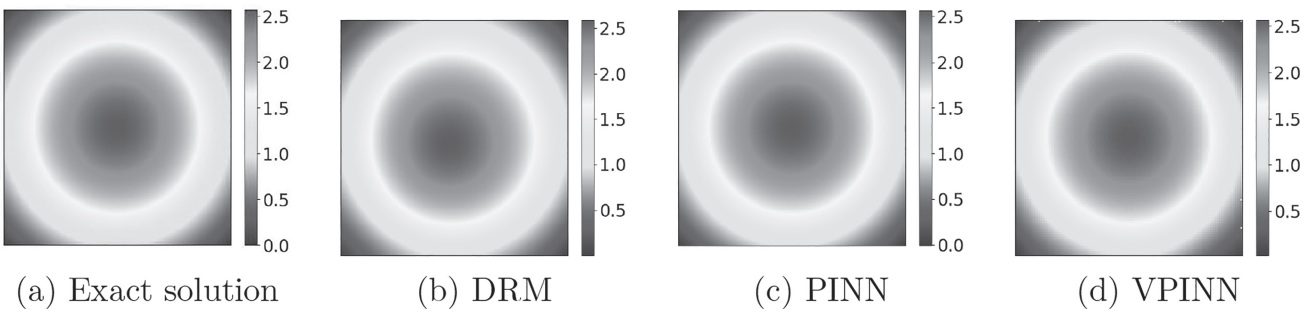


Fig. 2 Exact and approximate solutions for all the methods for Example 6.1.

Table 2 Error for all methods for Example 6.1

Method	$\|u-u_\theta\|_{L^2(\Omega)}$	L^2-relative error	$\|u-u_\theta\|_{H^1(\Omega)}$	H^1-relative error
DRM	0.026775	0.016579	0.257565	0.055632
PINN	0.000879	0.000545	0.013806	0.002982
VPINN	0.015616	0.009630	0.222021	0.047992

Table 3 Performance of Example 6.1 for various NN architecture

Methods	# Hidden Layers	# Neurons	# Collocation Points	$\boldsymbol{n}_L$	R	L^2 error	H^1 error	Training time (sec.)
DRM	2	60	$n=7000$ $m=2000$	3901	0.730045	0.040657	0.433367	114.69
	4	30	$n=5000$ $m=1000$	2911	0.599869	0.016590	0.157684	80.03
	6	20	$n=4000$ $m=1000$	2141	0.650495	0.032447	0.301550	63.19
PINN	2	60	$n=7000$ $m=2000$	3901	0.860108	0.001462	0.021980	170.08
	4	30	$n=5000$ $m=1000$	2911	0.609328	0.001210	0.017527	108.18
	6	20	$n=4000$ $m=1000$	2181	0.666157	0.001328	0.012010	97.96

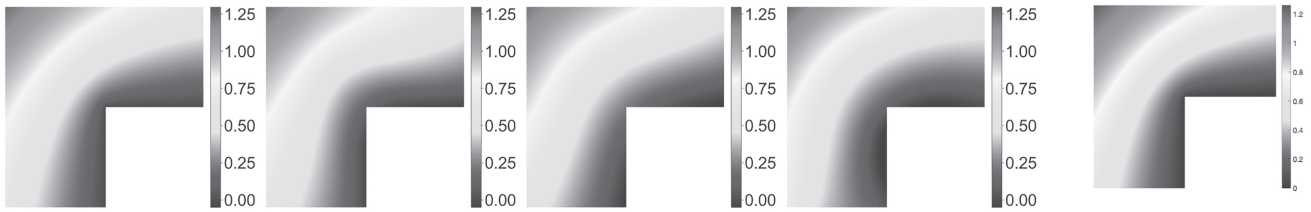
Remark 6.1 (comparison) We compare the theoretical estimates established in Theorems 3.1 and 4.1 with the practical performance observed in Example 6.1 for different neural network architectures. Table 3 illustrates the behavior of both DRM and PINN when using networks with 2, 4, and 6 hidden layers. From the table, it is evident that the DRM method is able to achieve an accuracy of order 10^{-2} using only a modest number of neurons, nonzero parameters $\boldsymbol{n}_L$, collocation points (n, m), and a relatively small parameter bound R. In contrast, the sufficient conditions of Theorem 3.1 predict the need for a significantly larger network architecture to guarantee the same accuracy. A similar discrepancy is also observed for the PINN method. These observations suggest that the current theoretical bounds on $\boldsymbol{n}_L$, m, n, and R in Theorems 3.1 and 4.1 are sufficient conditions but not necessary for the examples considered in this article. Note that in PINN only L^2 estimates are available theoretically; however we demonstrate both L^2 and H^1 error in numerical experiments.

Example 6.2 (DRM, PINN, VPINN, and adaptive FEMs on L-shape domain in 2D)

This example demonstrates the performance of DRM, PINN, VPINN, and the adaptive finite element method for a singular solution in an L-shape domain. The article [42, Example 4.2] makes a comparative study of neural network-based methods; however, in this article, we incorporate the performance of the adaptive finite element method. For the reproduction of the results of DRM, PINN, and VPINN, we use the same neural network architecture and parameter settings as in [42, Example 4.2]. For adaptive finite element methods, we use a mesh with 2366 nodes and 4522 elements.

Consider the Poisson equation (1.1) in the L-shaped domain $\Omega = (-1, 1)^2 \backslash [0, 1) \times (-1, 0]$. The source is given by $f = -\Delta u = 0$. This PDE has the exact solution $u = r^{\frac{2}{3}} \sin(\frac{2}{3}\theta)$ and the boundary data $g := u$ on $\partial\Omega$ [42, Example 4.2]. The exact and approximate solutions for all the three methods are displayed in Figure 3. The L^2 and H^1-errors and relative errors are shown in Table 4.

The current example highlights the well-known challenge of resolving corner singularities on L-shape domain. The results in Table 4 show that DRM and VPINN outperform standard PINN, with VPINN providing the most



(a) Exact (b) DRM (c) PINN (d) VPINN (e) AFEM

Fig. 3 Exact and approximate solutions for Example 6.2

Table 4 Error for all methods for Example 6.2.

Method	$\|u - u_\theta\|_{L^2(\Omega)}$	L^2-relative error	$\|u - u_\theta\|_{H^1(\Omega)}$	H^1-relative error
DRM	0.038890	0.064513	0.198077	0.200904
PINN	0.096221	0.159620	0.339566	0.344411
VPINN	0.037798	0.037798	0.148392	0.150510
AFEM	0.000313	0.000301	0.018300	0.010700

accurate approximation among the neural network-based approaches. Nevertheless, for this example, we observe that adaptive finite element methods (AFEM) perform better in comparison to VPINN. AFEM achieve errors that are several orders of magnitude smaller. To be more specific, AFEM yields a solution with about 99.97% accuracy, DRM achieves 93.55% accuracy, PINN 84.04%, and VPINN 96.23%. These results suggest that incorporating adaptivity techniques will potentially enhance the accuracy of DRM, PINN, and VPINN. The training time (in seconds) for DRM, PINN, VPINN, and AFEM for this example are 41, 58.01, 78.43, and 4.25, respectively.

Example 6.3 (**DRM and PINN in higher dimensions**)

Consider the Poisson problem (1.1) in six dimensional domain $\Omega = [0, 1]^6$. For the (known) exact solution $u = \sin(3\pi x_1)\sin(3\pi x_2)\sin(\pi x_3)\sin(\pi x_4) + \cos(\pi x_5) + \cos(\pi x_6)$, the source and boundary data are computed from (1.1).

We test the performance of DRM and PINN in this example. We use a DNN architecture with 4 hidden layers and 80 neurons in each layer. The domain data of size $n = 10000$ is uniformly sampled from the domain, and boundary data of size $m = 2000$ is uniformly sampled from the boundary. For both the methods, the boundary weight is taken to be 100, selected by trial and error. We use 10000 ADAM iterations to optimize the neural network parameters, with initial learning rate 10^{-3}, multiplied by 0.1 at 5000-th and 7000-th iterations. For DRM, the L^2 error is 0.181340 and L^2 relative error is 17.5%. For PINN, the L^2 error is 0.077701 and L^2 relative error is 7.5%, thus illustrating that PINN fares better for this example.

A Bounds for the approximation error

We recall the following result on the bound for the approximation error [27, Proposition 4.8] and [29, Theorem 4.1, Remark 4.1].

Proposition A.1 *(approximation error bound) Let $p \geq 1, m, k, d \in \mathbb{N}^+, m \geq k + 1$. Let ρ be the hyperbolic tangent or sigmoid function. For any $\delta > 0$ and $y \in W^{m,p}([0, 1]^d)$ with $\|y\|_{W^{m,p}([0,1]^d)} \leq 1$, there exists a neural network y_θ with depth $L = c\log(d + m)$ and non-zero weights $\mathbf{n}_L = c(d, m, p, k)\delta^{-\frac{d}{m-k-\mu}}$ such that*

$$\|y - y_\theta\|_{W^{k,p}([0,1]^d)} \leq \delta,$$

where μ is an arbitrarily small positive number. Moreover, the weights and biases θ in the neural network y_θ are bounded in absolute value by $R = c(d, m, p, k)\delta^{-2-\frac{2(\frac{d}{p}+d+k+\mu)+\frac{d}{p}+d}{m-k-\mu}}$.

B Abstract statistical error estimates

Split the statistical error as $\mathcal{E}_{stat} := \mathcal{E}^1_{stat} + \mathcal{E}^2_{stat}$ with $\mathcal{E}^1_{stat} = \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\sup_{u_\theta\in\mathcal{U}}[\mathcal{L}_\beta(u_\theta) - \widehat{\mathcal{L}_\beta}(u_\theta)]]$ and $\mathcal{E}^2_{stat} = \mathbb{E}_{\{\{X_k\}_{k=1}^n,\{Y_k\}_{k=1}^m\}}[\sup_{u_\theta\in\mathcal{U}}[\widehat{\mathcal{L}_\beta}(u_\theta) - \mathcal{L}_\beta(u_\theta)]]$. We only estimate $\mathcal{E}^1_{stat}$, and the esimate of $\mathcal{E}^2_{stat}$ follows similarly. Now the loss function has an interior part and a boundary part. Let the function classes $\mathcal{F}_i = \{h_i(u_\theta;\cdot) : \Omega \to \mathbb{R} |\, u_\theta \in \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)\}$ and $\mathcal{F}_b = \{h_b(u_\theta;\cdot) : \partial\Omega \to \mathbb{R} |\, u_\theta \in \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)\}$ be such that the continuous loss can be represented as

$$\mathcal{L}_\beta(u_\theta) := \int_\Omega h_i(u_\theta; x)dx + \beta\int_{\partial\Omega} h_b(u_\theta; s)ds. \tag{B.1}$$

The empirical loss reads

$$\widehat{\mathcal{L}_\beta}(u_\theta) := \frac{|\Omega|}{n}\sum_{k=1}^n h_i(u_\theta; X_k) + \beta\frac{|\partial\Omega|}{m}\sum_{j=1}^m h_b(u_\theta; Y_j). \tag{B.2}$$

We also define $\Delta\mathcal{E}_i = \sup_{h_i\in\mathcal{F}_i} |\Omega|\mathbb{E}_{U(\Omega)}h_i(u_\theta; X) - \frac{|\Omega|}{n}\sum_{k=1}^n h_i(u_\theta; X_k)$ and $\Delta\mathcal{E}_b = \sup_{h_b\in\mathcal{F}_b} |\partial\Omega|\mathbb{E}_{U(\partial\Omega)}h_b(u_\theta; Y) - \frac{|\partial\Omega|}{m}\sum_{j=1}^m h_b(u_\theta; Y_j)$. A similar bound holds for $\mathcal{E}^2_{stat}$ as well and we have

$$\mathcal{E}^1_{stat} \le \mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] + \beta\mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b] \text{ and } \mathcal{E}^2_{stat} \le \mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] + \beta\mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b]. \tag{B.3}$$

Now we prove bounds on $\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i]$ and $\mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b]$ using Rademacher complexity [2]. Rademacher complexity measures the complexity of a collection of functions by the correlation between function values with Rademacher random variables.

Definition B.1 (Rademacher complexity) Let $\mathcal{F}$ be a real-valued function class defined in the domain Ω (or on the boundary $\partial\Omega$) and $\xi = \{\xi_j\}_{j=1}^n$ (or $\xi = \{\xi_k\}_{k=1}^m$) be i.i.d. samples from the distribution $\mathcal{U}(\Omega)$ (or the distribution $\mathcal{U}(\partial\Omega)$). Then the Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ (or $\mathcal{R}_m(\mathcal{F})$) of the function class $\mathcal{F}$ is defined by

$$\mathcal{R}_n(\mathcal{F}) := \mathbb{E}_{\xi,\omega}\left[\sup_{v\in\mathcal{F}}\frac{1}{n}\sum_{j=1}^n \omega_j v(\xi_j)\right] \quad \left(\text{or } \mathcal{R}_m(\mathcal{F}) := \mathbb{E}_{\xi,\omega}\left[\sup_{v\in\mathcal{F}}\frac{1}{m}\sum_{k=1}^m \omega_k v(\xi_k)\right]\right),$$

where $\omega = \{\omega_j\}_{j=1}^l$ (with $l = n$ or m) are i.i.d Rademacher random variables with probability $P(\omega_j = 1) = P(\omega_j = -1) = \frac{1}{2}$.

The hypotheses **(H1)** and **(H2)** stated in Lemma 3.3 on boundedness and Lipschitz continuity of the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$ are useful to obtain an upper bound on Rademacher complexity.

The following lemma from [17, Lemma A.2, A.3, and A.4] is useful in determining M_i, M_b, Λ_i, and Λ_b from **(H1)**-**(H2)** in applications.

Lemma B.1 *For any $u := u_\theta \in \mathcal{U} = \mathcal{N}_\rho(L, \boldsymbol{n}_L, R)$, and $\tilde{u} := u_{\tilde{\theta}} \in \mathcal{U}$, with the activation function ρ as the hyperbolic tangent or sigmoid. The function class $\mathcal{U}$ consists of functions defined on Ω. The following estimates hold: for any $1 \le p \le d$, and $1 \le q \le n_\ell$ and for all $x \in \Omega$,*

$$|u(x)| \le (n_{L-1}+1)R, \tag{B.4}$$

$$|u_q^{(\ell)}(x) - \tilde{u}_q^{(\ell)}(x)| \le \begin{cases} \left(\prod_{k=1}^{\ell-1} n_k\right) R^{\ell-1} \sum_{j=1}^{\boldsymbol{n}_\ell} |\theta_j - \tilde{\theta}_j|, \quad \ell = 1, \dots, L-1, \\ \sqrt{\boldsymbol{n}_L} \left(\prod_{k=1}^{L-1} n_k\right) R^{L-1} \|\theta - \tilde{\theta}\|_{\ell^2}, \quad \ell = L, \end{cases} \tag{B.5}$$

$$|\partial_{x_p} u^{(\ell)}(x)| \le \left(\prod_{k=1}^{\ell-1} n_k\right) R^\ell, \quad \ell = 1, \dots, L, \tag{B.6}$$

$$|\partial_{x_p} u_q^{(\ell)}(x) - \partial_{x_p} \tilde{u}_q^{(\ell)}(x)| \le (\ell+1) \left(\prod_{k=1}^{\ell-1} n_k\right)^2 R^{2\ell-1} \sum_{j=1}^{\boldsymbol{n}_\ell} |\theta_j - \tilde{\theta}_j|, \quad \ell = 1, \dots, L. \tag{B.7}$$

$$|\partial_{x_p^2}^2 u_q^{(\ell)}(x)| \le \ell \left(\prod_{k=1}^{\ell-1} n_k\right)^2 R^{2\ell}, \quad \ell = 1, 2, \dots, L \tag{B.8}$$

$$|\partial_{x_p^2}^2 u(x) - \partial_{x_p^2}^2 \tilde{u}(x)| \le 2(L-1)L\eta\sqrt{\boldsymbol{n}_L} \left(\prod_{k=1}^{L-1} n_k\right)^3 R^{3L-3} \|\theta - \tilde{\theta}\|_{\ell^2}, \tag{B.9}$$

with $\eta = 1$ for sigmoid and $\eta = 2$ for hyperbolic tangent function in (B.9).

By a standard symmetrization argument, we have the following bounds on $\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i]$ and $\mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b]$ in terms of the Rademacher complexity of the sets $\mathcal{F}_i$ and $\mathcal{F}_b$. These bounds play a crucial role in deriving the final error estimate. The following estimates are derived using the ideas from [29, Lemma 5.3]; however, we provide a proof for completeness of this article.

Lemma B.2 (bounds for $\mathbb{E}_{\{X_k\}_{k=1}^n}(\Delta\mathcal{E}_i)$ and $\mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b]$). *With the notations above, it holds that*

$$\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] \le 2\mathcal{R}_n(\mathcal{F}_i) \text{ and } \mathbb{E}_{\{Y_k\}_{k=1}^m}[\Delta\mathcal{E}_b] \le 2\mathcal{R}_m(\mathcal{F}_b).$$

Proof We prove the first inequality here the second one follows similarly. Let $\{\tilde{X}_k\}_{k=1}^n$ be an independent copy of $\{X_k\}_{k=1}^n$. Then we have

$$\begin{aligned} |\Omega|\mathbb{E}_{U(\Omega)}[h_i(u_\theta; X)] - \frac{|\Omega|}{n}\sum_{k=1}^n h_i(u_\theta; X_k) &= \frac{|\Omega|}{n}\mathbb{E}_{\{\tilde{X}_k\}_{k=1}^n} \sum_{k=1}^n h_i(u_\theta; \tilde{X}_k) - \frac{|\Omega|}{n}\sum_{k=1}^n h_i(u_\theta; X_k) \\ &= \frac{|\Omega|}{n}\mathbb{E}_{\{\tilde{X}_k\}_{k=1}^n} \sum_{k=1}^n \Big[h_i(u_\theta; \tilde{X}_k) - h_i(u_\theta; X_k)\Big]. \end{aligned}$$

Taking supremum in $h_i(u_\theta; \cdot) \in \mathcal{F}_i$, applying Jensen's inequality [28], and taking expectation with respect to $\{X_k\}_{k=1}^n$ lead to

$$\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] \le \frac{|\Omega|}{n}\mathbb{E}_{\{X_k, \tilde{X}_k\}_{k=1}^n} \sup_{h\in\mathcal{F}_i} \sum_{k=1}^n \Big[h_i(u_\theta; \tilde{X}_k) - h_i(u_\theta; X_k)\Big].$$

By the independence of $\{X_k\}_{k=1}^n$ and $\{\tilde{X}_k\}_{k=1}^n$, inserting the Rademacher random variables σ_k does not change the distribution and hence

$$\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] \leq \frac{|\Omega|}{n}\mathbb{E}_{\{X_k,\tilde{X}_k,\sigma_k\}_{k=1}^n}\sup_{h\in\mathcal{F}_i}\sum_{k=1}^n \sigma_k\Big[h(u_\theta;\tilde{X}_k) - h_i(u_\theta;X_k)\Big].$$

Since $h_i(u_\theta;X_k)$ and $h_i(u_\theta;\tilde{X}_k)$ are independent, $\sigma_k h_i(u_\theta;\tilde{X}_k)$ and $-\sigma_k h_i(u_\theta;X_k)$ have the same distribution, and thus we have

$$\begin{aligned}\mathbb{E}_{\{X_k\}_{k=1}^n}[\Delta\mathcal{E}_i] &\leq \frac{|\Omega|}{n}\mathbb{E}_{\{\tilde{X}_k,\sigma_k\}_{k=1}^n}\sup_{h\in\mathcal{F}_i}\sum_{k=1}^n \sigma_k h_i(u_\theta;\tilde{X}_k) + \frac{|\Omega|}{n}\mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^n}\sup_{h\in\mathcal{F}_i}\sum_{k=1}^n -\sigma_k h_i(u_\theta;X_k)\\ &\leq \frac{2|\Omega|}{n}\mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^n}\sup_{h\in\mathcal{F}_i}\sum_{k=1}^n \sigma_k h_i(u_\theta;X_k) \leq 2|\Omega|\mathbb{E}_{\{X_k,\sigma_k\}_{k=1}^n}\Big[\sup_{h\in\mathcal{F}_i}\frac{1}{n}\sum_{k=1}^n \sigma_k h_i(u_\theta;X_k)\Big] \leq 2|\Omega|\mathcal{R}_n(\mathcal{F}_i).\end{aligned}$$

This completes the proof of the lemma. □

Now we bound the Rademacher complexities $\mathcal{R}_n(\mathcal{F}_i)$ and $\mathcal{R}_n(\mathcal{F}_b)$ of the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$, respectively. This is achieved by the Dudley integral formula, which is based on the concept of ε-covering number of function classes. Thus, we briefly describe the ε-covering number of a function class. Let $\mathcal{F}$ be a real-valued function class equipped with the metric ρ. An ε-cover of the class $\mathcal{F}$ with respect to the metric ρ is a collection of points $\{f_i\}_{i=1}^n \subset \mathcal{F}$ such that for every $f \in \mathcal{F}$, there exists at least one $i \in \{1,\ldots,n\}$ such that $\rho(f,f_i) \leq \varepsilon$. The ε-covering number $\mathcal{C}(\mathcal{F},\rho,\varepsilon)$ is the minimum cardinality among all ε-covers of the class $\mathcal{F}$ with respect to the metric ρ. Now we can state the well-known Dudley's theorem [45, Theorem 1.19].

Lemma B.3 *(Dudley's theorem) Let $M_{\mathcal{F}} := \sup_{f\in\mathcal{F}}\|f\|_{L^\infty(\Omega)}$, and $\mathcal{C}(\mathcal{F},\|\cdot\|_{L^\infty(\Omega)},\varepsilon)$ be the covering number of the set $\mathcal{F}$. Then the Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ is bounded by*

$$\mathcal{R}_n(\mathcal{F}) \leq \inf_{0<s<M_{\mathcal{F}}}\left(4s \;+\; 12n^{-\frac{1}{2}}\int_s^{M_{\mathcal{F}}}\left(\log\mathcal{C}(\mathcal{F},\|\cdot\|_{L^\infty(\Omega)},\varepsilon)\right)^{\frac{1}{2}}\mathrm{d}\varepsilon\right).$$

With the help of Lemma B.3, (**H1**), (**H2**) stated in Lemma 3.3, and the estimate of the ε-covering number $\mathcal{C}(\mathcal{F},\rho,\varepsilon)$, we have the following estimate of a neural network function class.

Lemma B.4 *(bound for $\mathcal{R}_n(\mathcal{F}_i)$and $\mathcal{R}_m(\mathcal{F}_b)$) Let $\mathcal{F}_i = \{h_i(u_\theta;\cdot) : \Omega \to \mathbb{R} \mid u_\theta \in \mathcal{N}_\rho(L,\boldsymbol{n}_L,R)\}$, and $\mathcal{F}_b = \{h_b(u_\theta;\cdot) : \Omega \to \mathbb{R} \mid u_\theta \in \mathcal{N}_\rho(L,\boldsymbol{n}_L,R)\}$ denote function classes, where $\mathcal{N}_\rho(L,\boldsymbol{n}_L,R)$ is a neural network function class. Let n and m be the number of Monte Carlo sample points associated to the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$, M_i and M_b be the uniform bounds of the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$, respectively, and λ_i and λ_b be the Lipschitz constants of the function classes $\mathcal{F}_i$ and $\mathcal{F}_b$, respectively. Then the bounds*

$$\mathcal{R}_n(\mathcal{F}_i) \leq cn^{-\frac{1}{4}}\mathbf{n}_L R^{\frac{1}{2}} M_i \Lambda_i^{\frac{1}{2}} \text{ and } \mathcal{R}_m(\mathcal{F}_b) \leq cm^{-\frac{1}{4}}\mathbf{n}_L R^{\frac{1}{2}} M_b \Lambda_b^{\frac{1}{2}},$$

hold, where the constant c depends on d.

Proof We prove the first estimate and the second one follows analogously. Recall that for any $\mathfrak{m} \in \mathbb{N}$, $r \in [1,\infty)$, $\varepsilon \in (0,1)$, and $B_r := \{x \in \mathbb{R}^{\mathfrak{m}} : \|x\|_{\ell^2} \leq r\}$, by counting argument (see, e.g., [15, Proposition 5] or [29, Lemma 5.4]), we have $\log\mathcal{C}(B_r,\|\cdot\|_{\ell^2},\varepsilon) \leq \mathfrak{m}\log(2r\sqrt{\mathfrak{m}}\varepsilon^{-1})$. It follows directly from the Lipschitz continuity of neural network functions and the estimate $\|\theta\|_{\ell^2} \leq \sqrt{\boldsymbol{n}_L}\|\theta\|_{\ell^\infty} \leq \sqrt{\boldsymbol{n}_L}R$ that

$$\log\mathcal{C}(\mathcal{F},\|\cdot\|_{L^\infty(\Omega)},\varepsilon) \leq \log\mathcal{C}(\mathcal{N}_Y,\|\cdot\|_{\ell^2},\Lambda_i^{-1}\varepsilon) \leq \boldsymbol{n}_L\log(2\boldsymbol{n}_L R\Lambda_i\varepsilon^{-1}),$$

where $\mathcal{N}$ denotes the parameter space for the function class $\mathcal{F}$, and Λ_i ,M_i are the Lipschitz and boundedness constant for $\mathcal{F}$. Then letting $s = n^{-\frac{1}{2}}$ in Lemma B.3 leads to

$$\mathcal{R}_n(\mathcal{F}) \leq 4n^{-\frac{1}{2}} + 12n^{-\frac{1}{2}} \int_{n^{-\frac{1}{2}}}^{M_i} \left(\boldsymbol{n}_L \log(2R\boldsymbol{n}_L \Lambda_i \varepsilon^{-1})\right)^{\frac{1}{2}} \mathrm{d}\varepsilon \leq 4n^{-\frac{1}{2}} + 12n^{-\frac{1}{2}} M_i \left(\boldsymbol{n}_L \log(2R\Lambda_i \boldsymbol{n}_L \sqrt{n})\right)^{\frac{1}{2}}$$

Since $1 \leq R, \mathbf{n}_L$ and $\log(x) \leq x$, $\mathcal{R}_n(\mathcal{F}) \lesssim n^{-\frac{1}{4}} \mathbf{n}_L R^{\frac{1}{2}} M_i \Lambda_i^{\frac{1}{2}}$ with the constant absorbed in $\lesssim$ depending on d. □

Lemmas B.2, B.4, and (B.3) establishes a bound for $\mathcal{E}_{stat}$ as

$$\mathcal{E}_{stat} := \mathcal{E}^1_{stat} + \mathcal{E}^2_{stat} \lesssim n^{-\frac{1}{4}} \mathbf{n}_L R^{\frac{1}{2}} M_i \Lambda_i^{\frac{1}{2}} + \beta m^{-\frac{1}{4}} \mathbf{n}_L R^{\frac{1}{2}} M_b \Lambda_b^{\frac{1}{2}} \tag{B.10}$$

where the constant absorbed in $\lesssim$ depends on d.

Acknowledgements The first author acknowledges the project ANRF/JBG/2025/000209/HAA. The second author acknowledges the financial support received from the Department of Science and Technology (DST), Government of India, under the INSPIRE Faculty Fellowship (DST/INSPIRE/04/2024/004854). The authors sincerely thank Prof. Thirupathi Gudi, Dr. Deepanshu Verma, and Shivani Mangal for their valuable comments that has greatly improved the quality of this article.

Competing interest The authors declare that there are no competing interests.

References

1. Thivin Anandh, Divij Ghose, Himanshu Jain, and Sashikumaar Ganesan. FastVPINNs: Tensor-Driven Acceleration of VPINNs for Complex Geometries. *SIAM Journal on Scientific Computing*, 47(3):C578–C600, 2025.
2. Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *J. Mach. Learn. Res.*, 3:463–482, 2002.
3. Atılım Güneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic Differentiation in Machine Learning: A Survey. *J. Mach. Learn. Res.*, 18:1–43, 2018.
4. Martin Berggren. Approximations of Very Weak Solutions to Boundary-Value Problems. *SIAM Journal on Numerical Analysis*, 42(2):860–877, 2004.
5. Stefano Berrone, Claudio Canuto, and Moreno Pintore. Solving PDEs by Variational Physics?Informed Neural Networks: An A Posteriori Error Analysis. *Annali dell'Università di Ferrara*, 68:575–595, 2022.
6. Stefano Berrone, Claudio Canuto, and Moreno Pintore. Variational Physics Informed Neural Networks: The Role of Quadratures and Test Functions. *Journal of Scientific Computing*, 92(3):100, 2022.
7. Stefano Berrone and Moreno Pintore. Meshfree Variational-Physics-Informed Neural Networks (MF-VPINN): An Adaptive Training Strategy. *Algorithms*, 17(9):415, 2024.
8. Patrick Billingsley. *Probability and Measure*. A Wiley-Interscience Publication. Wiley, New York, 3rd edition, 1995.
9. Susanne C. Brenner. Poincaré–Friedrichs Inequalities for Piecewise H^1 Functions. *SIAM Journal on Numerical Analysis*, 41(1):306–324, 2003.
10. Susanne C. Brenner and Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, 3rd edition, 2008.
11. Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
12. Adrian Celaya, David Fuentes, and Beatrice Riviere. An Adaptive Collocation Point Strategy for Physics Informed Neural Networks via the QR Discrete Empirical Interpolation Method. arXiv:2501.07700, 2025.
13. Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.
14. Matteo Cicuttin, Alexandre Ern, and Nicolas Pignet. *Hybrid High-Order Methods—A Primer with Applications to Solid Mechanics*. SpringerBriefs in Mathematics. Springer, Cham, 2021.
15. Felipe Cucker and Steve Smale. On the Mathematical Foundations of Learning. *Bull. Amer. Math. Soc. (N.S.)*, 39(1):1–49, 2002.
16. Lourenço Beirao da Veiga, Franco Brezzi, Andrea Cangiani, Gianmarco Manzini, Donatella Marini, and Alessandro Russo. Basic Principles of Virtual Element Methods. *Mathematical Models and Methods in Applied Sciences*, 23(1):199–214, 2013.

17. 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*, 51:31, 2025.
18. Daniele Antonio Di Pietro and Jérôme Droniou. *The Hybrid High-Order Method for Polytopal Meshes: Design, Analysis, and Applications*, volume 19 of *MS&A. Modeling, Simulation and Applications*. Springer, Cham, 2020.
19. John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.
20. Weinan E and Bing Yu. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Commun. Math. Stat.*, 6(1):1–12, 2018.
21. Alexandre Ern and Jean-Luc Guermond. *Finite Elements I: Approximation and Interpolation*, volume 72 of *Texts in Applied Mathematics*. Springer, Cham, 2021.
22. Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. *The Finite Volume Method*, volume VII. Elsevier, 2000.
23. Zhiwei Gao, Liang Yan, and Tao Zhou. Failure-Informed Adaptive Sampling for PINNs. *SIAM Journal on Scientific Computing*, 45(4):A1971–A1994, 2023.
24. B. Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of Lazy Training of Two-layers Neural Networks. *Advances in Neural Information Processing Systems*, pages 9108–9118, 2019.
25. Xavier Glorot and Yoshua Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
26. Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, and Carola-Bibiane Schönlieb. Can Physics-Informed Neural Networks Beat the Finite Element Method? *IMA J Appl Math.*, 89:143–174, 2024.
27. Ingo Gühring and Mones Raslan. Approximation Rates for Neural Networks With Encodable Weights in Smoothness Spaces. *Neural Networks*, 134:107–130, 2020.
28. Johan Jensen. Sur Les Fonctions Convexes Et Les Inégalités Entre Les Valeurs Moyennes. *Acta Mathematica*, 30:175–193, 1906.
29. Yuling Jiao, Yanming Lai, Yisu Lo, Yang Wang, and Yunfei Yang. Error Analysis of Deep Ritz Methods for Elliptic Equations. *Analysis and Applications*, 22(01):57–87, 2024.
30. Ehsan Kharazmi, Zhongqiang Zhang, and George E.M. Karniadakis. hp-VPINNs: Variational Physics-Informed Neural Networks with Domain Decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.
31. Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*, San Diego, 2015.
32. Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. 2021.
33. Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators. *Nature Machine Intelligence*, 3:218 – 229, 2019.
34. Lu Lu, Raphael Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G Johnson. Physics-Informed Neural Networks With Hard Constraints for Inverse Design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021.
35. Rongxin Lu, Jiwei Jia, Young Ju Lee, Zheng Lu, and Chensong Zhang. R-PINN: Recovery-Type A-Posteriori Estimator Enhanced Adaptive PINN. arXiv preprint arXiv:2506.10243, 2025.
36. Tao Luo and Haizhao Yang. *Chapter 11 - Two-layer Neural Networks for Partial Differential Equations: Optimization and Generalization Theory*. Numerical Analysis Meets Machine Learning. Elsevier, 2024.
37. Johannes Müller and Marius Zeinhofer. Error Estimates for the Deep Ritz Method with Boundary Penalty. *Proceedings of Machine Learning Research*, 190:215–230, 2022.
38. Aayushman Raina, Satyadev Badireddi, and Srinivasan Natesan. Application of PINN to Obtain Solution of Boundary Layer Problems Arising in Fluid Dynamics. *Mathematical Foundations of Computing*, 2025.
39. 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.
40. Chengping Rao, Hao Sun, and Yang Liu. Physics-Informed Deep Learning for Computational Elastodynamics Without Labeled Data. *Journal of Engineering Mechanics*, 147(8):04021043, 2021.
41. Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Boston, 11th edition, 2014.
42. Ramesh Chandra Sau and Luowei Yin. A Review of Neural Network Solvers for Second-Order Boundary Value Problems. arXiv:2407.00442v2, 2024.
43. John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics, Philadelphia, 2nd edition, 2004.
44. Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.
45. Michael M. Wolf. *Mathematical Foundations of Supervised Learning*. 2023. Graue Literatur.

46. Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A Comprehensive Study of Non-Adaptive and Residual-Based Adaptive Sampling for Physics-Informed Neural Networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.
47. 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.
48. Shaojie Zeng, Zong Zhang, and Qingsong Zou. Adaptive Deep Neural Networks Methods for High-Dimensional Partial Differential Equations. *Journal of Computational Physics*, 463:111232, 2022.

Neela Nataraj is a Professor in the Department of Mathematics at the Indian Institute of Technology Bombay. She obtained her Ph.D. from the Indian Institute of Technology Delhi in 1998. Her research focuses on numerical analysis, finite element methods, and scientific computing for partial differential equations. She is a Fellow of the Indian National Science Academy, Indian Academy of Sciences, the National Academy of Sciences, India, and is a recipient of the J. C. Bose Grant. She serves on the editorial boards of several international journals, that include *ESAIM: Mathematical Modelling and Numerical Analysis* and *Computational Methods in Applied Mathematics*. A recipient of the INSA Teachers Award, she has made significant contributions to promoting women's participation in mathematics in India through her leadership as Chair of the Executive Committee of the Indian Women and Mathematics and her service as a member of the Committee for Women in Mathematics of the International Mathematical Union.

Ramesh Chandra Sau is an Assistant Professor at the Indian Institute of Information Technology Raichur, Karnataka. He obtained his Ph.D. in Mathematics from the Indian Institute of Science, Bangalore in 2021. His research interests include numerical analysis, finite element methods, optimal control problems, and deep learning approaches for partial differential equations. He is a recipient of the *DST INSPIRE Faculty Fellowship (2025–2030)* awarded by the Department of Science and Technology, Government of India.

Authors and Affiliations

Neela Nataraj[1] · Ramesh Chandra Sau[2]

✉ Neela Nataraj
neela@math.iitb.ac.in

Ramesh Chandra Sau
rcsau@iiitr.ac.in; rcsau1994@gmail.com

[1] Department of Mathematics, Indian Institute of Technology, Bombay, India

[2] Indian Institute of Information Technology, Raichur, India