Convergence of Physics-Informed Neural Networks Applied to Linear Second-Order Elliptic Interface Problems

Sidi Wu[1,2], Aiqing Zhu[1,2], Yifa Tang[1,2] and Benzhuo Lu[1,2,*]

[1] *State Key Laboratory of Scientific and Engineering Computing, National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.*
[2] *School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.*

Abstract. With the remarkable empirical success of neural networks across diverse scientific disciplines, rigorous error and convergence analysis are also being developed and enriched. However, there has been little theoretical work focusing on neural networks in solving interface problems. In this paper, we perform a convergence analysis of physics-informed neural networks (PINNs) for solving second-order elliptic interface problems. Specifically, we consider PINNs with domain decomposition technologies and introduce gradient-enhanced strategies on the interfaces to deal with boundary and interface jump conditions. It is shown that the neural network sequence obtained by minimizing a Lipschitz regularized loss function converges to the unique solution to the interface problem in H^2 as the number of samples increases. Numerical experiments are provided to demonstrate our theoretical analysis.

AMS subject classifications: 65N12, 82B24, 68T07

Key words: Elliptic interface problems, generalization errors, convergence analysis, neural networks.

1 Introduction

Deep learning in the form of deep neural networks (DNNs) has been effectively used in diverse scientific disciplines beyond its traditional applications. In particular, thanks to their potential nonlinear approximation power [1–3], DNNs are being exploited to construct alternative approaches for solving partial differential equations (PDEs), e.g., the

---

*Corresponding author. *Email addresses:* `bzlu@lsec.cc.ac.cn` (B. Lu), `wsd@lsec.cc.ac.cn` (S. Wu), `zaq@lsec.cc.ac.cn` (A. Zhu), `tyf@lsec.cc.ac.cn` (Y. Tang)

deep Ritz method (DRM) [4] and physics-informed neural networks (PINNs) [5]. The key idea of these methods is to reformulate the solution to a PDE with a closed-form expression in the form of a neural network, the parameters of which are obtained by minimizing a physics-informed loss given by the corresponding PDE. The original works on the use of neural networks to solve PDEs were proposed in the 1990*s* [6,7], and this idea has recently been revisited with the renaissance of neural networks and the development of deep learning techniques; see e.g., [8–12] and references therein.

Elliptic interface problems are a widespread class of problems in scientific computing with many applications across diverse fields; see e.g. [13–16]. There are many accurate and efficient numerical methods in the literature for interface problems, such as the finite element method (FEM) [17,18], the discontinuous Galerkin method (DG) [19,20], the immersed interface method (IIM) [21,22], the immersed boundary method (IBM) [23], the boundary element method (BEM) [14], and the Voronoi interface method (VIM) [24]. In the last few decades, the numerical methods for solving interface problems have reached a certain maturity and made satisfactory progress. However, the above-mentioned methods usually require either a body-fitted or unfitted mesh to treat the interface problems, and the main difficulty lies in the body-fitted mesh generation or in the technique designed to dissect the intersecting geometry of the interface and properly discretize interface conditions. Interface problems are still challenging due to the low global regularity and irregular geometry of interfaces.

In recent years, many efforts have been made to use neural networks to solve interface problems since these methods are meshfree and can take advantage of deep learning techniques such as automatic differentiation and GPU acceleration. In particular, neural network-based approaches exhibit notable advantages in treating high-dimensional problems, inverse problems, and simultaneously solving parametric PDE problems that involve learning the solution operator (operator learning), which issues also exist in interface problems. In addition, the use of multiple neural networks based on the domain decomposition method (DDM) has attracted increasing attention as they are more accurate and flexible in dealing with the interface and have shown remarkable success in various interface problems [25–28]. This idea is further studied from the numerical aspect in our previous work [28], where the proposed interfaced neural networks are able to balance the interplay between different terms in the composite loss function and improve the performance in terms of accuracy and robustness. The above-mentioned works focus on obtaining empirical results, whereas we focus on theoretical aspects such as the convergence of PINNs for solving interface problems in this paper.

Along with the remarkable empirical achievements of deep learning methods, rigorous error and convergence analysis are also being developed and enriched. In previous work [29], the Hölder continuity constant was used to obtain the generalization analysis of PINNs in the case of linear second-order elliptic and parabolic type PDEs. [30,31] used quadrature points in the formulation of the loss and carried out an a-posteriori-type generalization error analysis of PINNs for both forward and inverse problems. [32] studied linear PDEs and proved both a priori and posterior estimates for PINNs and variational

PINNs in Sobolev spaces. [33] provided a theoretical understanding of the generalization abilities of PINNs and Extended PINNs (XPINNs) [25]. [34] derived an a priori generalization estimate for a class of second-order linear PDEs in the context of two-layer neural networks by assuming that the exact solutions of PDEs belong to a Barron-type space [35]. [36] provided a nonasymptotic convergence rate of PINNs with ReLU[3] networks for the second linear elliptic equation. For high-dimensional PDEs, [37] derived a priori and dimension explicit generalization error estimates for the DRM under the assumption that the solutions of the PDEs lie in the spectral Barron space, and [38] provided an analysis of the generalization error for linear Kolmogorov equations by using tools from statistical learning theory and covering number estimates of neural network hypothesis classes.

However, the majority of existing theoretical works are limited to differential equations with continuous coefficients; much less is known about the convergence of PINNs in solving interface problems, where the interaction at the interface introduces additional analytical challenges. In particular, it is reasonable to make the assumption that the network satisfies the boundary conditions for elliptic problems (see, for example, Theorem 3.4 in [29]), since there are several approaches [7, 39] to forcing neural networks to obey the boundary conditions intrinsically. But such approaches cannot be applied to interface jump conditions. When considering the convergence of interface problems, we are inevitably faced with the challenge of estimating errors caused by interface losses.

By extending the convergence results in [29] to elliptic interface problems, we provide a convergence theory for DDM-based PINNs to solve linear second-order elliptic interface problems. In this work, to deal with the error caused by the non-zero interface and boundary losses, we introduce a gradient enhancement strategy on interfaces inspired by [40], where the gradient information from the residual of the boundary and interface jump conditions is embedded into the loss function. Following the work of Shin *et al.* [29], we construct a specific Lipschitz regularization loss tailored for elliptic interface problems to quantify the generalization of PINN. Finally, we prove that the sequence of minimizers of the designed regularized loss function converges to the unique solution to the interface problem in H^2 under some reasonable assumptions. To the best of our knowledge, this is the first theoretical work that proves the convergence of neural network methods for solving elliptic interface problems. The main contributions of our work can be summarized as follows:

- We introduce gradient-enhanced strategies on interfaces to estimate the error caused by non-zero interface and boundary losses.

- We first provide the convergence analysis of PINNs in solving elliptic interface problems.

- We present several numerical experiments to validate the theoretical analysis.

The rest of this paper is organized as follows. In Section 2, some preliminaries, including notations and background knowledge of neural networks and interface problems, are

introduced. In Section 3, we briefly introduce the algorithm of PINN for solving elliptic interface problems and present the gradient-enhanced strategies on the interfaces. In Section 4, we present a convergence analysis, the proof of which is presented in Section 6. Numerical experiments are performed in Section 5 to validate the theoretical analysis. Finally, we conclude the paper in Section 7.

2 Preliminaries

2.1 Notations

We first introduce some notations. Let $\mathbf{x} = (x_1, \cdots, x_d)$ be a point in $\mathbb{R}^d$ ($d \geq 2$) and $\mathcal{U} \subset \mathbb{R}^d$ be an open set. Let $C(\mathcal{U}) = \{f : \mathcal{U} \to \mathbb{R}^{d'} | f \text{ is continuous}\}$ denotes the space of continuous functions. Let $\mathbb{Z}_+^d$ denotes the lattice of d-dimensional nonnegative integers. For $\mathbf{k} = (k_1, \cdots, k_d) \in \mathbb{Z}_+^d$, we set $|\mathbf{k}| := k_1 + \cdots + k_d$, and

$$D^{\mathbf{k}} = \frac{\partial^{|\mathbf{k}|}}{\partial x_1^{k_1} \cdots \partial x_d^{k_d}}.$$

For a positive integer k, we define

$$C^k(\mathcal{U}) := \{f : D^{\mathbf{k}} f \in C(\mathcal{U}) \text{ for all } |\mathbf{k}| \leq k\}.$$

Subsequently, we denote $[\mu]_{\mathcal{U}}$ to be the Lipshcitz constant of μ on $\mathcal{U}$, i.e.,

$$[\mu]_{\mathcal{U}} = \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{U}, \mathbf{x} \neq \mathbf{y}} \frac{\|\mu(\mathbf{x}) - \mu(\mathbf{y})\|_\infty}{\|\mathbf{x} - \mathbf{y}\|_\infty}.$$

In order to distinguish from the k-times continuously differentiable function space C^k, we denote $C^{k,L}(\mathcal{U})$ to be the collection of functions in C^k whose derivatives of order k are Lipschitz continuous.

Following [41], we present the definition of $H^s(E)$ on the boundary E. Here, we suppose E is a $(d-1)$-dimensional smooth manifold, i.e., there exists a collection of charts $\{(\mathcal{V}_i, \phi_i) \mid i \in I\}$ such that $\{\mathcal{V}_i\}_{i \in I}$ is a collection of open sets on E and covers E (i.e., $E = \cup_{i \in I} \mathcal{V}_i$), and such that ϕ_i is homeomorphism from $\mathcal{V}_i$ to an open subset $\mathcal{V}_i' := \phi_i(\mathcal{V}_i)$ of $\mathbb{R}^{d-1}$ for all $i \in I$ and the transition map $\phi_i \circ \phi_j^{-1} : \phi_j(\mathcal{V}_i \cap \mathcal{V}_j) \to \phi_i(\mathcal{V}_i \cap \mathcal{V}_j)$ is an infinitely differentiable mapping when $\mathcal{V}_i \cap \mathcal{V}_j \neq \varnothing$ for all $i, j \in I$. Let $\{\eta_i\}_{i \in I}$ be a partition of unity on E with compact support in $\mathcal{V}_i$ such that $\sum_i \eta_i(\mathbf{x}) = 1$ for all $\mathbf{x} \in E$ and η_i are infinitely differentiable. Then, if u is a function on E, we can decompose $u = \sum_i (\eta_i u)$, and define

$$\phi_i^*(\eta_i u)(\boldsymbol{\xi}) = (\eta_i u)(\phi_i^{-1}(\boldsymbol{\xi})), \quad \boldsymbol{\xi} \in \mathcal{V}_i'.$$

Finally, we define

$$H^s(E) = \{u \mid \phi_i^*(\eta_i u) \in H^s(\mathcal{V}_i'), \forall i \in I\},$$

with norm

$$\|u\|_{H^s(E)} = \left(\sum_i \|\phi_i^*(\eta_i u)\|_{H^s(\mathcal{V}_i')}^2 \right)^{\frac{1}{2}}. \tag{2.1}$$

It is easy to verify that $H^s(E)$ is a Hilbert space and that the different norms (2.1) are equivalent. We refer the readers to [41] for more details.

For given $\{\mathcal{V}_i, \phi_i, \eta_i\}_{i \in I}$ and $f \in C^2(E)$, we define that if dimension $d = 2$,

$$D_E f = \sum_i \frac{\partial \phi_i^*(\eta_i u)}{\partial \xi_1}, \quad D_E^2 f = \sum_i \frac{\partial^2 \phi_i^*(\eta_i u)}{\partial \xi_1^2},$$

and if dimension $d = 3$,

$$D_E f = \left(\sum_i \frac{\partial \phi_i^*(\eta_i u)}{\partial \xi_1}, \sum_i \frac{\partial \phi_i^*(\eta_i u)}{\partial \xi_2} \right), \quad D_E^2 f = \left(\sum_i \frac{\partial^2 \phi_i^*(\eta_i u)}{\partial \xi_1^2}, \sum_i \frac{\partial^2 \phi_i^*(\eta_i u)}{\partial \xi_1 \partial \xi_2}, \sum_i \frac{\partial^2 \phi_i^*(\eta_i u)}{\partial \xi_2^2} \right),$$

where ξ_j is the j-th component of ξ.

2.2 Neural networks

In addition, we introduce the employed network architecture, i.e., the feed-forward neural network (FNN), in this paper. Mathematically, an N-layer FNN is a nested composition of sequential linear functions and nonlinear activation functions, which takes the form

$$\mathbf{s}_i = f_i(\mathbf{s}_{i-1}) := \sigma(\mathbf{W}_i \mathbf{s}_{i-1} + \mathbf{b}_i), \text{ for } i = 1, \cdots, N-1,$$
$$\mathbf{s}_N = f_N(\mathbf{s}_{N-1}) := \mathbf{W}_N \mathbf{s}_{N-1} + \mathbf{b}_N,$$

where $\mathbf{s}_0 = \mathbf{x} \in \mathbb{R}^{d_{in}}$ is the input variable, $\mathbf{s}_i \in \mathbb{R}^{d_i}$ denotes the output of the i-th hidden layer, $\mathbf{s}_N \in \mathbb{R}^{d_{out}}$ is the corresponding output, and $\mathbf{W}_i \in \mathbb{R}^{d_{i+1} \times d_i}$ and $\mathbf{b}_i \in \mathbb{R}^{d_{i+1}}$ are trainable parameters. $\sigma : \mathbb{R} \to \mathbb{R}$ is the nonlinear activation function applied element-wise to a vector. Popular examples include the rectified linear unit (ReLU) $\text{ReLU}(z) = \max(0, z)$, the logistic sigmoid $\text{Sig}(z) = 1/(1 + e^{-z})$ and the hyperbolic tangent $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$. Equipped with those definitions, the FNN representation of a continuous function can be viewed as

$$\mathcal{NN}(\mathbf{x}) = f_N \circ \cdots \circ f_1(\mathbf{x}). \tag{2.2}$$

Furthermore, we denote all the trainable parameters (e.g., $\mathbf{W}_i$, $\mathbf{b}_i$) in (2.2) as $\boldsymbol{\theta} \in \Theta$, where $\boldsymbol{\theta}$ is a high-dimensional vector and Θ is the space of $\boldsymbol{\theta}$. Given a network architecture $\overrightarrow{\boldsymbol{n}}$ (e.g., the number of layers and the width of each hidden layer), we denote the set of all expressible functions (hypothesis space) as

$$\mathcal{H}_{\overrightarrow{\boldsymbol{n}}}^{\text{NN}} = \{ \mathcal{NN}(\cdot; \overrightarrow{\boldsymbol{n}}, \boldsymbol{\theta}) : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}} \,\big|\, \boldsymbol{\theta} \in \Theta \}. \tag{2.3}$$

2.3 Elliptic interface problems

Let $\Omega = \Omega_1 \cup \Omega_2$ be a bounded domain in $\mathbb{R}^d$ with smooth boundary $\partial\Omega$. Let $\Omega_1 \subset \Omega$ be an open domain with smooth boundary $\Gamma = \partial\Omega_1 \subset \Omega$ and $\Omega_2 = \Omega \backslash \Omega_1$ (see Fig. 1 for an illustration). We consider the following linear second-order elliptic interface problem

$$-\nabla \cdot (a_i \nabla u) + b_i u = f_i, \quad \text{in } \Omega_i, \ i = 1,2, \tag{2.4a}$$

$$[\![a\nabla u \cdot \mathbf{n}]\!] = \psi, \quad \text{on } \Gamma, \tag{2.4b}$$

$$[\![u]\!] = \varphi, \quad \text{on } \Gamma, \tag{2.4c}$$

$$u = g, \quad \text{on } \partial\Omega, \tag{2.4d}$$

where $[\![\mu]\!] := \mu|_{\Omega_2} - \mu|_{\Omega_1}$ denotes the jump of a quantity μ across Γ, and $\mathbf{n}$ denotes the unit outward normal of Ω_1. The coefficients

$$a(\mathbf{x}) = \begin{cases} a_1(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega_1, \\ a_2(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \quad b(\mathbf{x}) = \begin{cases} b_1(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega_1, \\ b_2(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega_2, \end{cases}$$

are piecewise spatial functions. The unknown part of this problem is the exact solution u^*, while others are given in advance.
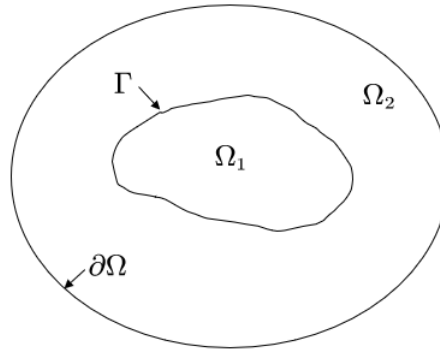

Figure 1: A schematic view of the geometry description.

3 Neural network methods for linear second-order elliptic interface problems

In this section, we present a brief overview of physics-informed neural networks (PINNs) [5] for solving linear second-order elliptic interface problems. Since the physical solutions to elliptic interface problems are usually non-smooth or even discontinuous across the interface, it is natural to use domain decomposition methods (DDMs) in the PINN framework [25–28]. In the context of DDM-based deep learning methods, the computational

domain is decomposed into several disjoint subdomains according to the interface, and the solution to the interface problem is the combination and ensemble of multiple local networks, where each of them is responsible for prediction in one subdomain.

Under the PINN framework, we approximate the latent solution to interface problems by two FNNs, i.e., $u_1(\mathbf{x},\boldsymbol{\theta}_1)\big|_{\Omega_1}$ and $u_2(\mathbf{x},\boldsymbol{\theta}_2)\big|_{\Omega_2}$. Let $\boldsymbol{\theta}=(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$ denotes all tunable parameters of the networks (e.g., weights and biases). We then use the constraints implied by Eq. (2.4) and the boundary and interface jump conditions to train the networks. Let us denote the number of training data points by $\boldsymbol{m}=(m_{r_1},m_{r_2},m_\Gamma,m_b)$, where m_{r_1},m_{r_2},m_Γ and m_b represent the number of training samples in Ω_1, Ω_2, Γ and $\partial\Omega$, respectively. Then, a physics-informed model can be trained by minimizing the following composite empirical loss function

$$\text{Loss}_{\boldsymbol{m}}^{\text{PINN}}(u_1,u_2;\boldsymbol{\lambda}):=\lambda_{r_1}\mathcal{M}_{\Omega_1}+\lambda_{r_2}\mathcal{M}_{\Omega_2}+\lambda_b\mathcal{M}_b+\lambda_{\Gamma_D}\mathcal{M}_{\Gamma_D}+\lambda_{\Gamma_N}\mathcal{M}_{\Gamma_N}, \quad (3.1)$$

where $\boldsymbol{\lambda}=(\lambda_{r_1},\lambda_{r_2},\lambda_\Gamma,\lambda_b)\geq 0$ (element-wise inequality), the loss terms $\mathcal{M}_{\Omega_1}$ and $\mathcal{M}_{\Omega_2}$ correspond to the PDE residuals (2.4a) in Ω_1 and Ω_2, $\mathcal{M}_b$, $\mathcal{M}_{\Gamma_D}$ and $\mathcal{M}_{\Gamma_N}$ enforce the boundary condition (2.4d), interface jump conditions (2.4c) and (2.4b), respectively. For a typical interface problem (2.4), we define

$$\mathcal{L}_i[u_i]=-\nabla\cdot(a_i\nabla u_i)+b_iu_i, \quad \text{with} \quad i=1,2, \quad \mathcal{B}[u_2]=u_2,$$
$$\mathcal{I}_D[u_1,u_2]=u_2-u_1, \quad \mathcal{I}_N[u_1,u_2]=a_2\nabla u_2\cdot\mathbf{n}-a_1\nabla u_1\cdot\mathbf{n},$$

which can be derived by automatic differentiation [42]. Then, the loss terms in $\text{Loss}_{\mathbf{m}}^{\text{PINN}}$ (3.1) would take the specific form

$$\mathcal{M}_{\Omega_1}=\frac{1}{m_{r_1}}\sum_{i=1}^{m_{r_1}}\left|\mathcal{L}_1[u_1](\mathbf{x}_{r_1}^i)-f_1(\mathbf{x}_{r_1}^i)\right|^2, \quad \mathcal{M}_{\Omega_2}=\frac{1}{m_{r_2}}\sum_{i=1}^{m_{r_2}}\left|\mathcal{L}_2[u_2](\mathbf{x}_{r_2}^i)-f_2(\mathbf{x}_{r_2}^i)\right|^2,$$

$$\mathcal{M}_b=\frac{1}{m_b}\sum_{i=1}^{m_b}\left\|\mathcal{B}[u_2](\mathbf{x}_b^i)-g(\mathbf{x}_b^i)\right\|_2^2, \quad \mathcal{M}_{\Gamma_D}=\frac{1}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma^i)-\varphi(\mathbf{x}_\Gamma^i)\right\|_2^2,$$

$$\mathcal{M}_{\Gamma_N}=\frac{1}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma^i)-\psi(\mathbf{x}_\Gamma^i)\right\|_2^2,$$

where $\{\mathbf{x}_b^i\}_{i=1}^{m_b}:=\mathcal{T}_b^{m_b}$ denotes the boundary data points, $\{\mathbf{x}_\Gamma^i\}_{i=1}^{m_\Gamma}:=\mathcal{T}_\Gamma^{m_\Gamma}$ denotes the interface data points, and $\{\mathbf{x}_{r_j}^i\}_{i=1}^{m_{r_j}}:=\mathcal{T}_{r_j}^{m_{r_j}}$ denotes the training data points that are randomly placed insider the subdomain Ω_j with $j=1,2$. Here, we suppose these four types of data sets are independently and identically (iid) sampled from probability distributions μ_{r_1}, μ_{r_2}, μ_Γ and μ_b, respectively.

In PINNs, we only enforce the residual of boundary and interface jump conditions to be zero, while in this work, we introduce gradient-enhanced strategies to the PINN framework to estimate the error caused by non-zero boundary and interface losses. Specifically, the high-order gradient information of the interface(s) is embedded into the loss

function by redefining the following loss terms,

$$\mathcal{M}_b = \frac{1}{m_b} \sum_{i=1}^{m_b} \left\| \mathcal{B}[u_2](\mathbf{x}_b^i) - g(\mathbf{x}_b^i) \right\|_2^2, \quad \mathcal{M}_{\Gamma_D} = \frac{1}{m_\Gamma} \sum_{i=1}^{m_\Gamma} \left\| \mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma^i) - \boldsymbol{\varphi}(\mathbf{x}_\Gamma^i) \right\|_2^2,$$

$$\mathcal{M}_{\Gamma_N} = \frac{1}{m_\Gamma} \sum_{i=1}^{m_\Gamma} \left\| \mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma^i) - \boldsymbol{\psi}(\mathbf{x}_\Gamma^i) \right\|_2^2,$$

where

$$\mathcal{B}[u_2] = (u_2, D_{\partial\Omega} u_2, D_{\partial\Omega}^2 u_2), \quad \mathcal{I}_D[u_1,u_2] = (u_2 - u_1, D_\Gamma(u_2 - u_1), D_\Gamma^2(u_2 - u_1)),$$
$$\mathcal{I}_N[u_1,u_2] = (a_2 \nabla u_2 \cdot \mathbf{n} - a_1 \nabla u_1 \cdot \mathbf{n}, D_\Gamma(a_2 \nabla u_2 \cdot \mathbf{n} - a_1 \nabla u_1 \cdot \mathbf{n})),$$

and

$$\boldsymbol{\varphi} = (\varphi, D_\Gamma \varphi, D_\Gamma^2 \varphi), \quad \boldsymbol{\psi} = (\psi, D_\Gamma \psi), \quad \mathbf{g} = (g, D_{\partial\Omega} g, D_{\partial\Omega}^2 g).$$

Note that since $u^*(\mathbf{x}) - g(\mathbf{x}) = 0$ for any $\mathbf{x}$ on the boundary, we know that for any positive integer k, the derivative $D_{\partial\Omega}^k(u^* - g)$ is zero. The same is true for the interface residuals. Hence, it is acceptable for us to enforce the derivatives of the residual of interface(s) to be zero. Obviously, the gradient-enhanced empirical PINN loss is an upper bound for vanilla empirical PINN loss. Unless otherwise stated, the rest of the paper discusses gradient-enhanced empirical PINN loss $\text{Loss}_{\mathbf{m}}^{\text{PINN}}$ (3.1). In addition, motivated by the upper bound [29], we consider the Lipschitz regularized loss function

$$\text{Loss}_m(u_1,u_2;\boldsymbol{\lambda},\boldsymbol{\lambda}^R) := \text{Loss}_{\mathbf{m}}^{\text{PINN}}(u_1,u_2;\boldsymbol{\lambda}) + \lambda_{r_1}^R R_{r_1}(u_1) + \lambda_{r_2}^R R_{r_2}(u_2)$$
$$+ \lambda_b^R R_b(u_2) + \lambda_{\Gamma_D}^R R_{\Gamma_D}(u_1,u_2) + \lambda_{\Gamma_N}^R R_{\Gamma_N}(u_1,u_2), \qquad (3.2)$$

where $\boldsymbol{\lambda}^R = (\lambda_{r_1}^R, \lambda_{r_2}^R, \lambda_\Gamma^R, \lambda_b^R) \geq 0$ (element-wise inequality), and R_{r_1}, R_{r_2}, R_b, R_{Γ_D}, R_{Γ_N} are regularization functionals. Specifically,

$$R_{r_1}(u_1) = \left[\mathcal{L}_1[u_1] \right]_{\Omega_1}^2, \quad R_{r_2}(u_2) = \left[\mathcal{L}_2[u_2] \right]_{\Omega_2}^2, \quad R_b(u_2) = \left[\mathcal{B}[u_2] \right]_{\partial\Omega}^2,$$
$$R_{\Gamma_D}(u_1,u_2) = \left[\mathcal{I}_D[u_1,u_2] \right]_\Gamma^2, \quad R_{\Gamma_N}(u_1,u_2) = \left[\mathcal{I}_N[u_1,u_2] \right]_\Gamma^2.$$

For the convenience of following analysis, we denote the expected loss of Eq. (2.4) (when $\boldsymbol{\lambda}^R = 0$) by $\text{Loss}^{\text{PINN}}(u_1,u_2;\boldsymbol{\lambda})$. More precisely,

$$\text{Loss}^{\text{PINN}}(u_1,u_2;\boldsymbol{\lambda}) = \lambda_{r_1} \left\| \mathcal{L}_1[u_1] - f_1 \right\|_{L^2(\Omega_1;\mu_{r_1})}^2 + \lambda_{r_2} \left\| \mathcal{L}_2[u_2] - f_2 \right\|_{L^2(\Omega_2;\mu_{r_2})}^2$$
$$+ \lambda_b \left\| \mathcal{B}[u_2] - g \right\|_{L^2(\partial\Omega;\mu_b)}^2 + \lambda_{\Gamma_D} \left\| \mathcal{I}_D[u_1,u_2] - \boldsymbol{\varphi} \right\|_{L^2(\Gamma;\mu_\Gamma)}^2$$
$$+ \lambda_{\Gamma_N} \left\| \mathcal{I}_N[u_1,u_2] - \boldsymbol{\psi} \right\|_{L^2(\Gamma;\mu_\Gamma)}^2. \qquad (3.3)$$

Remark 3.1. It is noted that the use of the gradient-enhanced strategy and Lipschitz regularization is for the convergence analysis. We will numerically verify that such technologies do not affect performance.

Remark 3.2. The present paper only considers the high regularity setting and that point-wise evaluations are well-defined. Specifically, it is required that $\mathcal{L}_i[u_i] \in C(\Omega_i)$ for $i=1,2$, $\mathcal{I}_D[u_1,u_2] \in C(\Gamma)$, $\mathcal{I}_N[u_1,u_2] \in C(\Gamma)$, and $\mathcal{B}[u_2] \in C(\partial\Omega)$ for all $(u_1,u_2) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m})$, and $f_i \in C(\Omega_i)$ for $i=1,2$, $\psi \in C(\Gamma)$, $\varphi \in C(\Gamma)$ and $g \in C(\partial\Omega)$.

4   Main results

We first present assumptions on the training data distributions based on the probability space filling arguments [43] to guarantee that random samples drawn from probability distributions can fill up both the interior of the domains Ω_1 and Ω_2 as well as the boundary $\partial\Omega$ and interface Γ.

Assumption 4.1 (Random sampling). For the interface problem (2.4), let μ_{r_1}, μ_{r_2}, μ_Γ and μ_b be probability distributions defined on Ω_1, Ω_2, Γ and $\partial\Omega$, respectively. Let $\rho_{r_1}(\rho_{r_2})$ be the probability density of $\mu_{r_1}(\mu_{r_2})$ with respect to d-dimensional Lebesgue measure on $\Omega_1(\Omega_2)$. Let $\rho_\Gamma(\rho_b)$ be the probability density of $\mu_\Gamma(\mu_b)$ with respect to the $(d-1)$-dimensional Hausdorff measure on $\Gamma(\partial\Omega)$.

1. ρ_{r_1}, ρ_{r_2}, ρ_Γ and ρ_b are supported on $\overline{\Omega}_1$, $\overline{\Omega}_2$, Γ and $\partial\Omega$, respectively. Also, $\inf_{\Omega_1}\rho_{r_1} > 0$, $\inf_{\Omega_2}\rho_{r_2} > 0$, $\inf_\Gamma\rho_\Gamma > 0$, and $\inf_{\partial\Omega}\rho_b > 0$.

2. For $\epsilon > 0$, there exists partitions of Ω_1, Ω_2, Γ and $\partial\Omega$, $\{\Omega_{1,j}^\epsilon\}_{j=1}^{K_{r_1}}$, $\{\Omega_{2,j}^\epsilon\}_{j=1}^{K_{r_2}}$, $\{\Gamma_j^\epsilon\}_{j=1}^{K_\Gamma}$ and $\{\partial\Omega_j^\epsilon\}_{j=1}^{K_b}$ that depend on ϵ such that for each j, there are cubes $H_\epsilon(\mathbf{z}_j^{r_1})$, $H_\epsilon(\mathbf{z}_j^{r_2})$, $H_\epsilon(\mathbf{z}_j^\Gamma)$ and $H_\epsilon(\mathbf{z}_j^b)$ of side length ϵ centered at $\mathbf{z}_j^{r_1} \in \Omega_{1,j}^\epsilon$, $\mathbf{z}_j^{r_2} \in \Omega_{2,j}^\epsilon$, $\mathbf{z}_j^\Gamma \in \Gamma_j^\epsilon$ and $\mathbf{z}_j^b \in \partial\Omega_j^\epsilon$, respectively, satisfying $\Omega_{1,j}^\epsilon \subset H_\epsilon(\mathbf{z}_j^{r_1})$, $\Omega_{2,j}^\epsilon \subset H_\epsilon(\mathbf{z}_j^{r_2})$, $\Gamma_j^\epsilon \subset H_\epsilon(\mathbf{z}_j^\Gamma)$ and $\partial\Omega_j^\epsilon \subset H_\epsilon(\mathbf{z}_j^b)$.

3. There exists positive constants $c_{r_1}, c_{r_2}, c_\Gamma, c_b$ such that $\forall \epsilon > 0$, the partitions from the above satisfy $c_{r_1}\epsilon^d \leq \mu_{r_1}(\Omega_{1,j}^\epsilon)$, $c_{r_2}\epsilon^d \leq \mu_{r_2}(\Omega_{2,j}^\epsilon)$, $c_\Gamma\epsilon^{d-1} \leq \mu_\Gamma(\Gamma_j^\epsilon)$ and $c_b\epsilon^{d-1} \leq \mu_b(\partial\Omega_j^\epsilon)$ for all j.

   There exists positive constants $C_{r_1}, C_{r_2}, C_\Gamma, C_b$ such that for $\forall \mathbf{x}_{r_1} \in \Omega_1$, $\mathbf{x}_{r_2} \in \Omega_2$, $\mathbf{x}_\Gamma \in \Gamma$ and $\mathbf{x}_b \in \partial\Omega$, we have $\mu_{r_1}(B_\epsilon(\mathbf{x}_{r_1}) \cap \Omega_1) \leq C_{r_1}\epsilon^d$, $\mu_{r_2}(B_\epsilon(\mathbf{x}_{r_2}) \cap \Omega_2) \leq C_{r_2}\epsilon^d$, $\mu_\Gamma(B_\epsilon(\mathbf{x}_\Gamma) \cap \Gamma) \leq C_\Gamma\epsilon^{d-1}$ and $\mu_b(B_\epsilon(\mathbf{x}_b) \cap \partial\Omega) \leq C_b\epsilon^{d-1}$ where $B_\epsilon(x)$ is a closed ball of radius ϵ centered at x.

   Here C_{r_1}, c_{r_1} depend only on (Ω_1, μ_{r_1}), C_{r_2}, c_{r_2} depend only on (Ω_2, μ_{r_2}), C_Γ, c_Γ depend only on (Γ, μ_Γ) and C_b, c_b depend only on $(\partial\Omega, \mu_b)$.

In contrast to the traditional applications of deep learning, such as image classification and natural language processing, where the data distributions are unknown and data sampling is very expensive, the aforementioned assumptions are mild and easy to satisfy when solving interface problems, as the computation domain and interface are given and the data distribution is known (e.g., the uniform probability distribution).

In addition, for the loss function (3.2) to be well-defined, we have to make some assumptions about the interface problem (2.4) and the hypothesis space of neural networks. Here, the network architecture $\overrightarrow{n}$ is expected to grow proportionally to the number of training samples m, thus we rewrite $\mathcal{H}_{\overrightarrow{n}}^{NN}$ as $\mathcal{H}_m$ for simplicity.

Assumption 4.2 (Interface problem and hypothesis space). Let $\mathcal{H}_{1,m}$ and $\mathcal{H}_{2,m}$ be the class of neural networks defined on $\overline{\Omega}_1$ and $\overline{\Omega}_2$, respectively.

1. Let $f_1 \in C^{0,L}(\Omega_1)$, $f_2 \in C^{0,L}(\Omega_2)$, $\psi \in C^{1,L}(\Gamma)$, $\varphi \in C^{2,L}(\Gamma)$ and $g \in C^{2,L}(\partial\Omega)$.

2. For each m, $\mathcal{H}_{1,m} \subset C^{2,L}(\overline{\Omega}_1)$, $\mathcal{H}_{2,m} \subset C^{2,L}(\overline{\Omega}_2)$ such that for any $(u_1, u_2) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m})$, $\mathcal{L}_1[u_1] \in C^{0,L}(\Omega_1)$, $\mathcal{L}_2[u_2] \in C^{0,L}(\Omega_2)$, $\mathcal{I}_{\Gamma_D}[u_1, u_2] \in C^{0,L}(\Gamma)$, $\mathcal{I}_{\Gamma_N}[u_1, u_2] \in C^{0,L}(\Gamma)$ and $\mathcal{B}[u_2] \in C^{0,L}(\Gamma)$.

3. For each m, $\mathcal{H}_{1,m}(\mathcal{H}_{2,m})$ contains a network $\hat{u}_{1,m}(\hat{u}_{2,m})$ satisfying

$$\text{Loss}_m^{PINN}(\hat{u}_{1,m}, \hat{u}_{2,m}; \lambda) = \mathcal{O}\left(\max\{m_{r_1}, m_{r_2}, m_\Gamma^{\frac{d}{d-1}}, m_b^{\frac{d}{d-1}}\}^{-\frac{1}{2}-\frac{1}{d}}\right),$$

4. and

$$\sup_m \left[\mathcal{L}_1[\hat{u}_{1,m}]\right]_{\Omega_1} < \infty, \quad \sup_m \left[\mathcal{L}_2[\hat{u}_{2,m}]\right]_{\Omega_2} < \infty, \quad \sup_m \left[\mathcal{B}[\hat{u}_{2,m}]\right]_{\partial\Omega} < \infty,$$

$$\sup_m \left[\mathcal{I}_{\Gamma_D}[\hat{u}_{1,m}, \hat{u}_{2,m}]\right]_\Gamma < \infty, \quad \sup_m \left[\mathcal{I}_{\Gamma_N}[\hat{u}_{1,m}, \hat{u}_{2,m}]\right]_\Gamma < \infty.$$

Popular activation functions, such as sigmoid $\text{Sig}(z)$ and $\tanh(z)$, could satisfy the Lipschitz condition. It is known that FNNs can simultaneously and uniformly approximate a continuous function and various of its partial derivatives [1, 44–48]. In particular, standard multi-layer FNNs with a tanh activation function are capable of approximating arbitrary functions from the Sobolev space, provided sufficiently many hidden units are available [46]. Thus, the third term in Assumption 4.2 can be attained.

With these assumptions, the main result is presented as follows.

Theorem 4.1. *Suppose Assumptions 4.1 and 4.2 hold. Let m_{r_1}, m_{r_2}, m_b and m_Γ be the number of iid samples from μ_{r_1}, μ_{r_2}, μ_b and μ_Γ, respectively, and $m_{r_2} = \mathcal{O}(m_{r_1})$, $m_\Gamma = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$, $m_b = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$. Let*

$$C_m = 3\max\{\kappa_{r_1}\sqrt{d}^d m_{r_1}^{\frac{1}{2}}, \kappa_{r_2}\sqrt{d}^d m_{r_2}^{\frac{1}{2}}, \kappa_b \sqrt{d}^{d-1} m_b^{\frac{1}{2}}, \kappa_\Gamma \sqrt{d}^{d-1} m_\Gamma^{\frac{1}{2}}\},$$

where $\kappa_{r_1} = \frac{C_{r_1}}{c_{r_1}}$, $\kappa_{r_2} = \frac{C_{r_2}}{c_{r_2}}$, $\kappa_b = \frac{C_b}{c_b}$, $\kappa_\Gamma = \frac{C_\Gamma}{c_\Gamma}$. Let $\hat{\lambda}_m^R = (\hat{\lambda}_{r_1,m}^R, \hat{\lambda}_{r_2,m}^R, \hat{\lambda}_{b,m}^R, \hat{\lambda}_{\Gamma_D,m}^R, \hat{\lambda}_{\Gamma_N,m}^R)$ be a vector where

$$\hat{\lambda}_{r_1,m}^R = \frac{3\lambda_{r_1}dc_{r_1}^{-\frac{2}{d}}}{C_m} \cdot m_{r_1}^{-\frac{1}{d}}, \quad \hat{\lambda}_{r_2,m}^R = \frac{3\lambda_{r_2}dc_{r_2}^{-\frac{2}{d}}}{C_m} \cdot m_{r_2}^{-\frac{1}{d}}, \quad \hat{\lambda}_{b,m}^R = \frac{3\lambda_b dc_b^{-\frac{2}{d-1}}}{C_m} \cdot m_b^{-\frac{1}{d-1}},$$

$$\hat{\lambda}_{\Gamma_D,m}^R = \frac{3\lambda_{\Gamma_D}dc_\Gamma^{-\frac{2}{d-1}}}{C_m} \cdot m_\Gamma^{-\frac{1}{d-1}}, \quad \hat{\lambda}_{\Gamma_N,m}^R = \frac{3\lambda_{\Gamma_N}dc_\Gamma^{-\frac{2}{d-1}}}{C_m} \cdot m_\Gamma^{-\frac{1}{d-1}}.$$

Let λ_m^R be a vector satisfying

$$\lambda_m^R \geq \hat{\lambda}_m^R, \quad \left\|\lambda_m^R\right\|_\infty = \mathcal{O}(\left\|\hat{\lambda}_m^R\right\|_\infty).$$

Let $(u_{1,m}, u_{2,m}) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m})$ be a minimizer of the Lipschitz regularized loss $\text{Loss}_m(\cdot; \lambda, \lambda_m^R)$ (3.2). Then the following holds,

- *The interface problem (2.4) has a unique solution $u^* \in H^2(\Omega_1) \cap H^2(\Omega_2)$.*

- *With probability 1 over iid samples,*

$$\lim_{m_{r_1} \to \infty} u_{1,m} = u^* \text{ in } H^2(\Omega_1), \quad \lim_{m_{r_1} \to \infty} u_{2,m} = u^* \text{ in } H^2(\Omega_2).$$

Theorem 4.1 shows that the minimizers of the Lipschitz regularized empirical losses (3.2) converge to the unique solution to the interface problem (2.4) in H^2 as the number of samples increases. The proof is postponed to Section 6.

Remark 4.1. That $(u_{1,m}, u_{2,m}) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m})$ is a minimizer means

$$\text{Loss}_m(u_{1,m}, u_{2,m}; \lambda, \lambda_m^R) \leq \text{Loss}_m(u_1, u_2; \lambda, \lambda_m^R), \quad \text{for } \forall (u_1, u_2) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m}).$$

We remark that this condition can be relaxed to

$$\text{Loss}_m(u_{1,m}, u_{2,m}; \lambda, \lambda_m^R) \leq \text{Loss}_m(\hat{u}_{1,m}, \hat{u}_{2,m}; \lambda, \lambda_m^R), \text{ for } \hat{u}_{1,m}, \hat{u}_{2,m} \text{ given in Assumption 4.2.}$$

5   Numerical experiments

In this section, we present numerical evidence to verify our analysis. We limit ourselves to the idealized setting considered for the theoretical analysis and to two-dimensional (2D) interface problems for the sake of illustration. For simplicity, we refer to the results obtained by minimizing the empirical loss $\text{Loss}_m^{\text{PINN}}(u_1, u_2; \lambda = \mathbf{1})$ (3.1) without gradient enhancement on boundary and interface, i.e., the original PINN loss [5], as "PINN", to the results obtained by minimizing the gradient-enhanced PINN empirical loss (3.1) as "PINN-GE", and to the results obtained by minimizing the loss (3.2) as "LIPR-GE". Throughout all benchmarks, we show the L^2 and H^2 convergence of the trained neural networks obtained by LIPR-GE as the number of training data increases. Note that the values of $b_1 = b_2 = 0$ (in Eq. (2.4)) are used for all the test examples except Example 5.5, where the non-zero values are mentioned. All code and data accompanying this manuscript are publicly available at `https://github.com/bzlu-Group/ConvergencePINNInterface`.

5.1   Settings

Network architecture.   The feed-forward tanh-neural networks of depth 5 and width 200 are employed for all experiments.

Training data.   The training points are randomly drawn from the corresponding do-mains. Specially, taking $m_r = 10,30,50,100,300,\cdots,10000$, we randomly sample the train-ing data points $\{\mathbf{x}_r^i\}_{i=1}^{m_r}$ inside the domain Ω and then divided them into two parts, i.e., $\{\mathbf{x}_{r_1}^i\}_{i=1}^{m_{r_1}}$ and $\{\mathbf{x}_{r_2}^i\}_{i=1}^{m_{r_2}}$, according to the interface. In addition, taking $m_b = m_\Gamma = 10\lfloor m_r^{1/2} \rfloor$, we randomly sample the training data points $\{\mathbf{x}_b^i\}_{i=1}^{m_b}$ and $\{\mathbf{x}_\Gamma^i\}_{i=1}^{m_\Gamma}$ on the boundary and interface, respectively, see Fig. 2 (left) for an illustration. Note that this strategy satisfies the conditions stated in Theorem 4.1, i.e., $m_{r_2} = \mathcal{O}(m_{r_1})$, $m_\Gamma = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$, $m_b = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$.

Gradient enhancement.   The boundary and interface are parameterized by ϑ, i.e., $\mathbf{x} = (x_1(\vartheta), x_2(\vartheta))$, and the gradients of the functions defined on the boundary or interface are derived by the related parameterized functions.

Optimization.   We train the networks for 10,000 stochastic gradient descent steps by minimizing the loss using the Adam optimizer [49]. The initial learning rate is 1×10^{-3}, halved every 1000 iterations; and full-batch training is employed.

Regularization.   For the Lipschitz regularized terms in loss function (3.2), we use the maximum of the sup norm of the derivative over the set of training data points to estimate the Lipschitz constant, more precisely,

$$[\mathcal{L}_1[u_1]]_{\Omega_1}^2 = \max_{1\leq j\leq m_{r_1}} \left\| \nabla \mathcal{L}_1[u_1](\mathbf{x}_{r_1}^j) \right\|_\infty^2, \quad [\mathcal{L}_2[u_2]]_{\Omega_2}^2 = \max_{1\leq j\leq m_{r_2}} \left\| \nabla \mathcal{L}_2[u_2](\mathbf{x}_{r_2}^j) \right\|_\infty^2,$$

$$[\mathcal{B}[u_2]]_{\partial\Omega}^2 = \max_{1\leq j\leq m_b} \left\| \nabla \mathcal{B}[u_2](\mathbf{x}_b^j) \right\|_\infty^2, \quad [\mathcal{I}_D[u_1,u_2]]_\Gamma^2 = \max_{1\leq j\leq m_\Gamma} \left\| \nabla \mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma^j) \right\|_\infty^2,$$

$$[\mathcal{I}_N[u_1,u_2]]_\Gamma^2 = \max_{1\leq j\leq m_\Gamma} \left\| \nabla \mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma^j) \right\|_\infty^2.$$

The weights in loss function (3.2) are set as $\boldsymbol{\lambda} = (\lambda_{r_1}, \lambda_{r_2}, \lambda_b, \lambda_{\Gamma_D}, \lambda_{\Gamma_N}) = \mathbf{1}$, $\lambda_{r_1}^R = \lambda_{r_2}^R = \frac{1}{m_r}$, and $\lambda_{\Gamma_D}^R = \lambda_{\Gamma_N}^R = \frac{1}{m_\Gamma \sqrt{m_r}}$, $\lambda_b^R = \frac{1}{m_b \sqrt{m_r}}$, which satisfy the conditions stated in Theorem 4.1.

Test.   After training, the L^2 error between the reference solution u^* and the obtained neural network solution $\hat{u}$ is measured as

$$\varepsilon_{L^2} = \|u - \hat{u}\|_{L^2} := \sqrt{\frac{1}{N}\sum_{i=1}^{N} |u^*(\mathbf{x}_i) - \hat{u}(\mathbf{x}_i)|^2},$$

where N denotes the total number of test points in the computational domain, see Fig. 2 (right) for an illustration. The H^2 error between u^* and $\hat{u}$ is measured as

$$\varepsilon_{H^2} = \left\{ \sum_{|\boldsymbol{\tau}|\leq 2} \|D^{\boldsymbol{\tau}}(u^* - \hat{u})\|_{L^2}^2 \right\}^{\frac{1}{2}}.$$
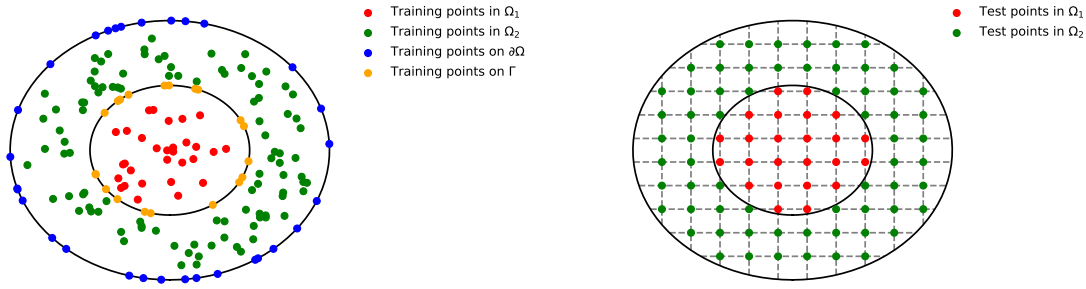
Figure 2: An illustration of the train and test data points. Left: These dots represent the training data points, which are randomly sampled in related regions. Right: An illustration of the equidistant test data points in the computational domain.

5.2 An elliptic interface problem with constant coefficients

In this case, we consider Eq. (2.4) with a circle interface, which is given as $(x_1(\vartheta), x_2(\vartheta)) = (1 + \cos(\vartheta), 1 + \sin(\vartheta))$, where $\vartheta \in [0, 2\pi]$. The computational domain in this problem is a closed disk with a radius of two and centered at $(1,1)$. The discontinuous coefficient a is given as

$$a(x_1, x_2) = \begin{cases} 1, & \text{in } \Omega_1, \\ 2, & \text{in } \Omega_2. \end{cases}$$

The exact solution to this equation is given by

$$u^*(x_1, x_2) = \begin{cases} 2\tanh(x_1 + x_2), & \text{in } \Omega_1, \\ \tanh(x_1 + x_2), & \text{in } \Omega_2. \end{cases}$$

Note that this solution can be exactly represented by the neural network we employed in this case. The corresponding source term is $f(x_1, x_2) = 8\tanh(x_1 + x_2) - 8\tanh(x_1 + x_2)^3$ and the boundary and jump conditions can be found by using the exact solution.

In Fig. 3, we compare the numerical solution obtained by LIPR-GE with that obtained by PINN. It can be seen from this figure that both two numerical solutions have small L^2 and H^2 errors, while the result of LIPR-GE performs better than that of PINN. Numerical results indicate that PINN with gradient enhancement is acceptable in practice as it does not affect the performance of vanilla PINN.

In Fig. 4, we show the L^2 and the H^2 errors of the results obtained by LIPR-GE with respect to the number of interior points m_r. Note that the number of points on the boundary and on the interface increases as m_r increases. To show the convergence trend, we construct a univariate linear regression, i.e., $\log_{10}\varepsilon = \alpha \log_{10} m_r + \beta$, for the logarithm of the numerical solution error $\log_{10}\varepsilon$ (i.e., $\log_{10}\varepsilon_{L^2}$ or $\log_{10}\varepsilon_{H^2}$) versus $\log_{10} m_r$, and estimate the parameters α and β using the linear least square algorithm. The dash lines here are the results of the regression. As expected by Theorem 4.1, the L^2 and H^2 errors decrease as m_r increases, implying the L^2- and H^2-convergence.
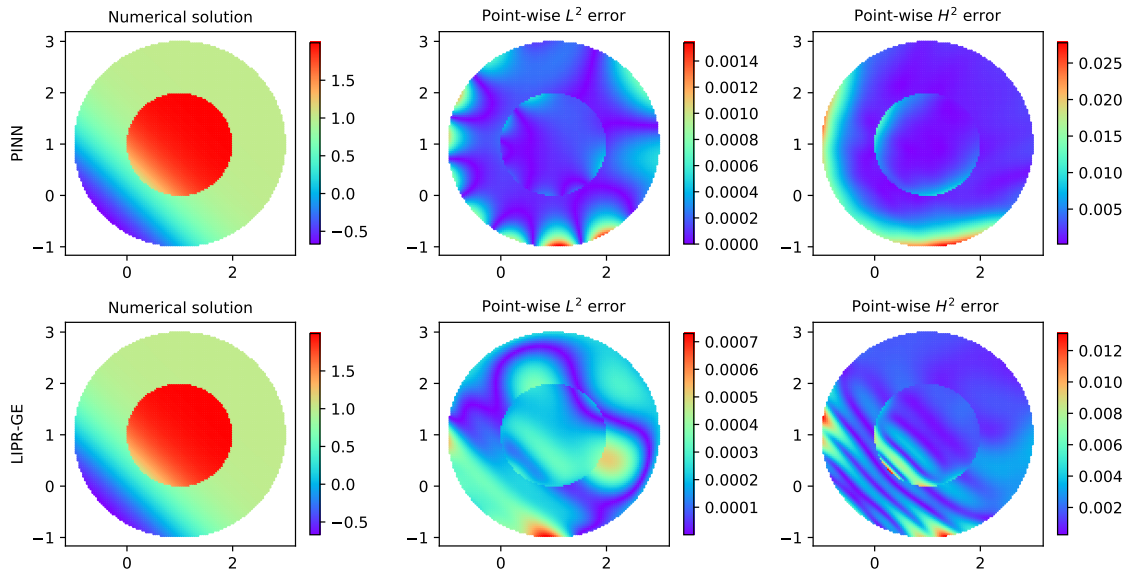
Figure 3: The numerical solution and point-wise errors for Example 5.2. The first row gives the numerical results of PINN whereas the second row gives that of LIPR-GE. Here, the number of training data points $m_r = 10000$.
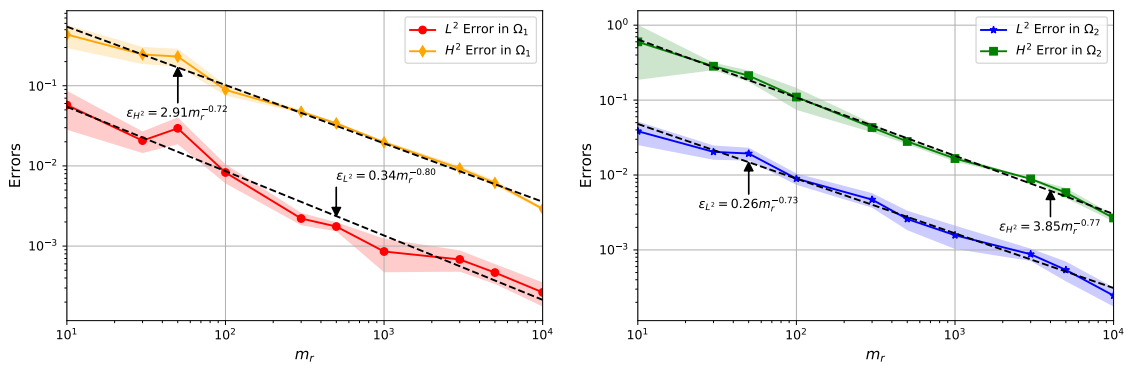

Figure 4: The L^2 and H^2 convergence of the errors of Example 5.2 with respect to the number of training data points. The shaded regions are the one-standard-derivation from five runs with different training data and network initialization. Here, the number of test data points in Ω_1 and Ω_2 are $N_1 = 7833$ and $N_2 = 23584$, respectively.

5.3 An elliptic interface problem with high contrast coefficients

Next, we consider Eq. (2.4) in the case of a large contrast in discontinuous coefficient a. Here, the computational domain Ω is a disk with a radius of one, centered at the origin. The interface is defined as $(x_1, x_2) = (r_0 \cos(\vartheta), r_0 \sin(\vartheta))$, where $r_0 = 0.5$. This exact
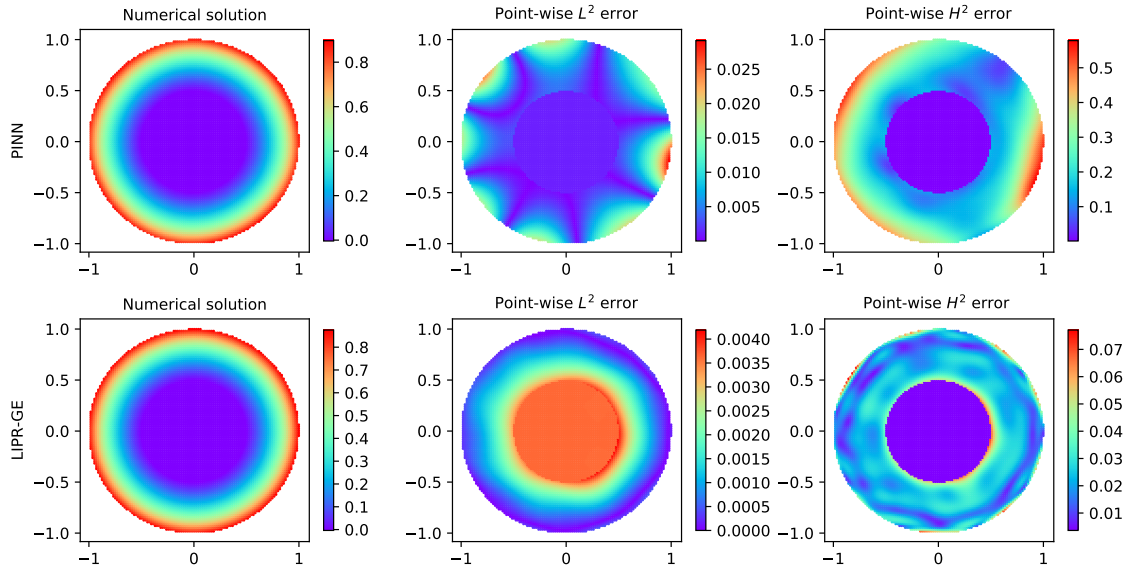
Figure 5: The numerical solution and point-wise errors for Example 5.3. The first row gives the numerical results of PINN whereas the second row gives that of LIPR-GE. Here, the number of training data points $m_r = 10000$.

solution [50] (in the polar coordinate) of this example is expressed as

$$u^*(r,\theta) = \begin{cases} \dfrac{r^3}{1000}, & r < r_0, \\ r^3 - \dfrac{999}{1000}r_0^3, & r \ge r_0, \end{cases}$$

where $r = \sqrt{x_1^2 + x_2^2}$ and the discontinuous coefficient is stated as

$$a(x_1, x_2) = \begin{cases} 1000, & r < r_0, \\ 1, & r \ge r_0. \end{cases}$$

Source terms, boundary and interface jump conditions are calculated from the above exact solution.

We first investigate the effect of gradient-enhanced strategies on the interfaces and depict the numerical results obtained by LIPR-GE and those obtained by PINN in Fig. 5. It can be observed that the auxiliary loss terms do not affect performance but also significantly reduce the absolute point-wise error at the interface and boundary. In addition, we continue testing the convergence. In Fig. 6, we show the L^2 and H^2 errors obtained by LIPR-GE with respect to the number of training data. We see that the rate of convergence is at least $\mathcal{O}(m_r^{-0.50})$. The results in this figure clearly demonstrate the convergence trend of L^2-error and H^2-error.
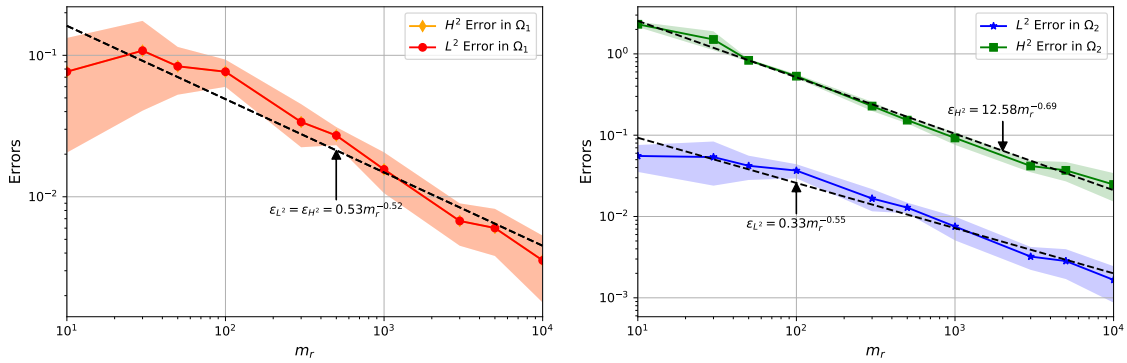
Figure 6: The L^2 and H^2 convergence of the errors of Example 5.3 with respect to the number of training data points. The shaded regions are the one-standard-derivation from five runs with different training data and network initialization. Here, the number of test data points in Ω_1 and Ω_2 are $N_1 = 7829$ and $N_2 = 23588$, respectively.

5.4 An elliptic interface problem with variable coefficients

In this case, we consider Eq. (2.4) with variable coefficients. Here, the computational domain is a closed disk placed at the origin with a radius of two, and the interface is circular with a radius of one and centered at the origin. The interface points can be obtained via $(x_1, x_2) = (\cos(\vartheta), \sin(\vartheta))$, where $\vartheta \in [0, 2\pi)$. The coefficient a is defined to be

$$a(x_1, x_2) = \begin{cases} \cos(x_1 + x_2) + 2, & \text{in } \Omega_1, \\ \sin(x_1 + x_2) + 2, & \text{in } \Omega_2. \end{cases}$$

The exact solution to this problem is given by [22]

$$u^*(x_1, x_2) = \begin{cases} \sin(x_1 + x_2), & \text{in } \Omega_1, \\ \ln(x_1^2 + x_2^2), & \text{in } \Omega_2, \end{cases}$$

and the corresponding source term is

$$f(x_1, x_2) = \begin{cases} 4(\cos(x_1 + x_2) + 1)\sin(x_1 + x_2), & \text{in } \Omega_1, \\ -2\cos(x_1 + x_2)\dfrac{x_1 + x_2}{x_1^2 + x_2^2}, & \text{in } \Omega_2. \end{cases}$$

Numerical results for Example 5.4 are displayed in Fig. 7 and Fig. 8. In Fig. 7, we first present a comparison between the exact and the numerical solution obtained using PINN or LIPR-GE. It can be observed that the solution obtained by LIPR-GE is in good agreement with that of PINN. As expected, the accuracy of LIPR-GE is not affected by the gradient enhancement and Lipschitz regularization. Furthermore, Fig. 8 shows the L^2- and H^2- convergence for neural network solutions in subdomains Ω_1 (left) and Ω_2 (right). Again, we observe that the errors in both subdomains decreases rapidly as the number of training data points m_r increases. The theoretical results in Theorem 4.1 are still valid in solving elliptic interface problems with variable coefficients.
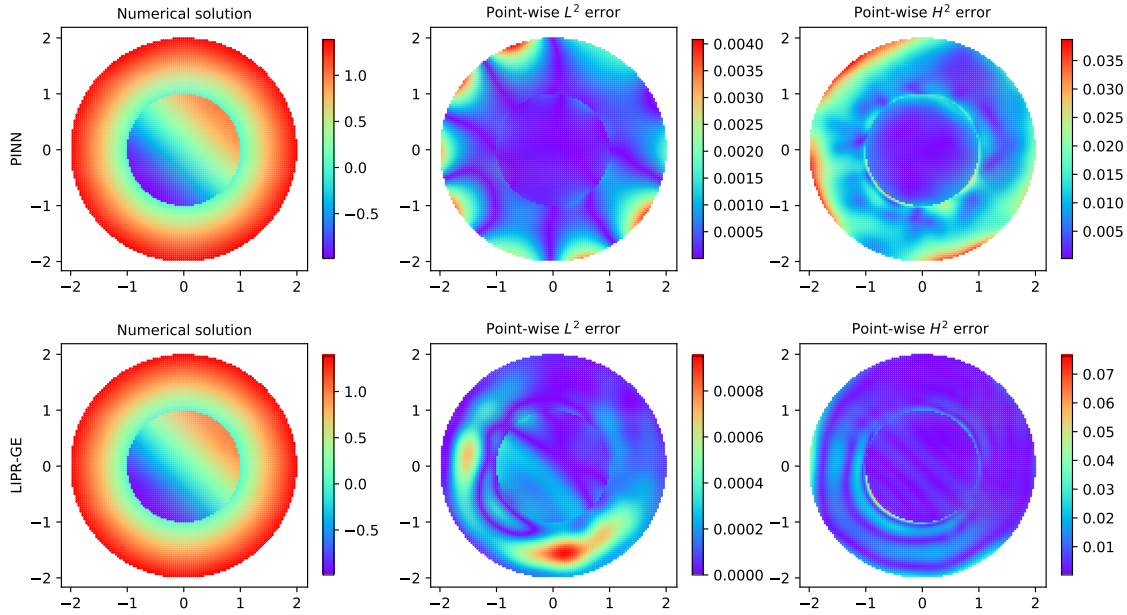
Figure 7: The numerical solution and point-wise errors for Example 5.4. The first row gives the numerical results of PINN whereas the second row gives that of LIPR-GE. Here, the number of training data points $m_r = 10000$.
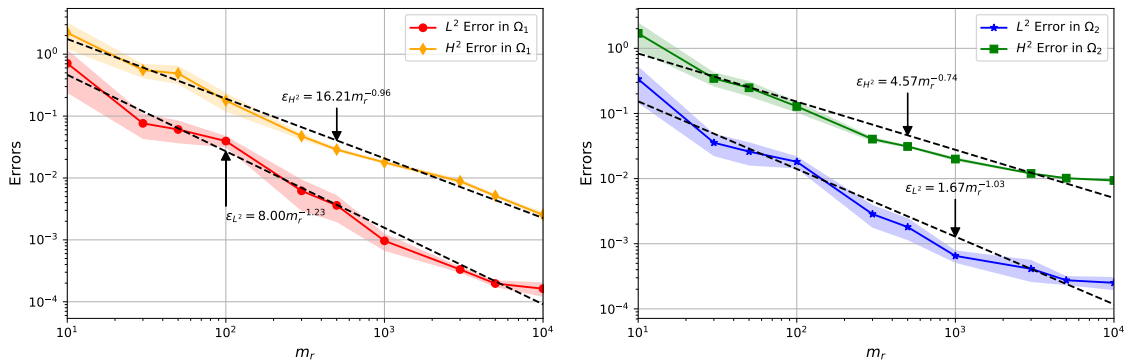

Figure 8: The L^2 and H^2 convergence of the errors of Example 5.4 with respect to the number of training data points. The shaded regions are the one-standard-derivation from five runs with different training data and network initialization. Here, the number of test data points in Ω_1 and Ω_2 are $N_1 = 7829$ and $N_2 = 23556$, respectively.

5.5   An elliptic interface problem with irregular geometry

In this case, we consider Eq. (2.4) with a complicated interface Γ (see Fig. 9 left), which consists of both convex and concave curves and is expressed with the following parametric equations

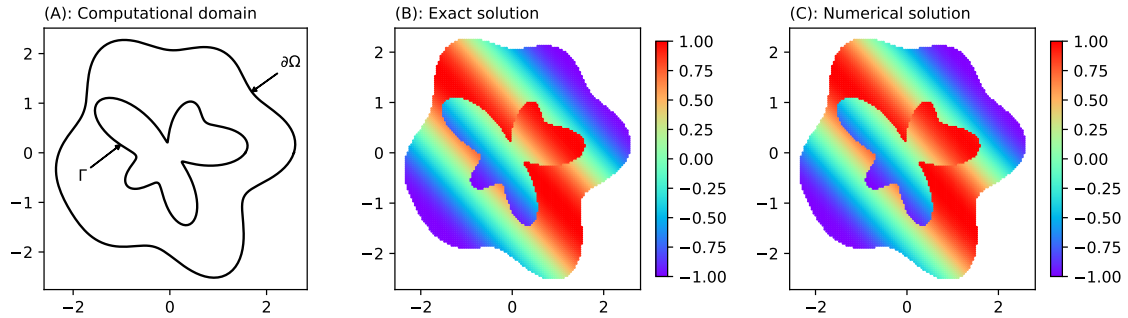$$x_1(\vartheta) = r\cos(\vartheta), \quad x_2(\vartheta) = r\sin(\vartheta),$$

Figure 9: The computational domain, exact and numerical solutions for Example 5.5. Here, the number of training data points $m_r = 10000$.

where $r = 1 + 0.36\sin(3\vartheta) + 0.16\cos(2\vartheta) + 0.4\cos(5\vartheta)$, $\vartheta \in [0, 2\pi]$. Computational domain is shown in Fig. 9. The boundary points (in polar coordinates) are obtained as $x_1 = 1.5r\cos(\vartheta)$ and $x_2 = 1.5r\sin(\vartheta)$, where $r = 1.5 + 0.14\sin(4\vartheta) + 0.12\cos(6\vartheta) + 0.09\cos(5\vartheta)$, $\vartheta \in [0, 2\pi)$. The coefficient $b_1 = b_2 = -1$, and the discontinuous coefficient a is defined to be

$$a(x, y) = \begin{cases} x_1 x_2, & \text{in } \Omega_1, \\ x_1^2 + x_2^2, & \text{in } \Omega_2. \end{cases}$$

The exact solution is set to be

$$u^*(x_1, x_2) = \begin{cases} \sin(x_1 + x_2), & \text{in } \Omega_1, \\ \cos(x_1 + x_2), & \text{in } \Omega_2. \end{cases}$$

The necessary source terms, boundary and interface jump conditions can be derived from this exact solution.

The computational domain, the exact solution of Example (5.5), and the numerical solution obtained by LIPR-GE are shown in Fig. 9. And the point-wise L^2 and H^2 errors in the whole domain for PINN, PINN-GE, and LIPR-GE are presented in Fig. 10. It is observed that LIPR-GE is less accurate than Vanilla PINN and PINN-GE. One explanation is that the additional auxiliary loss terms, especially the Lipschitz regularization, make it difficult for the optimization to find the minimizer. However, our following results clearly demonstrate that LIPR-GE can still recover the exact solution up to $\mathcal{O}(10^{-3})$ accuracy in L^2 and $\mathcal{O}(10^{-2})$ accuracy in H^2. Fig. 11 summarizes the convergent evolution of the L^2 and H^2 errors obtained by LIPR-GE with respect to the number of training data points. Clearly, the numerical results demonstrate both the L^2 and H^2-convergence of the errors, which are consistent to the theoretical analysis of this paper.
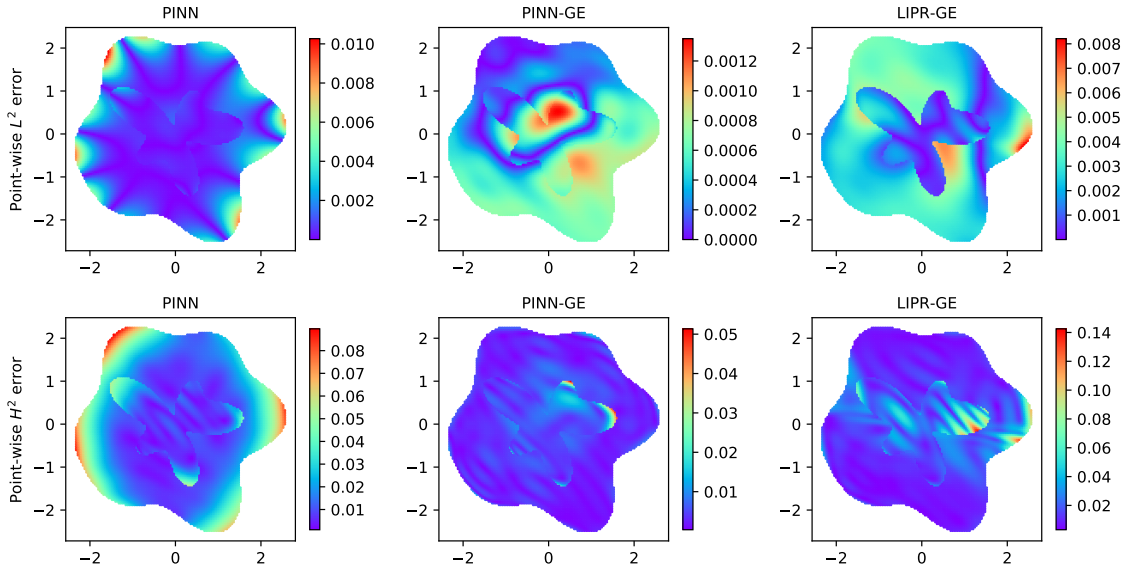
Figure 10: Point-wise errors for Example 5.5. The errors of PINN, PINN-GE, and LIPR-GE are placed in the first, second, and third columns, respectively. Here, the number of training data points $m_r = 10000$.
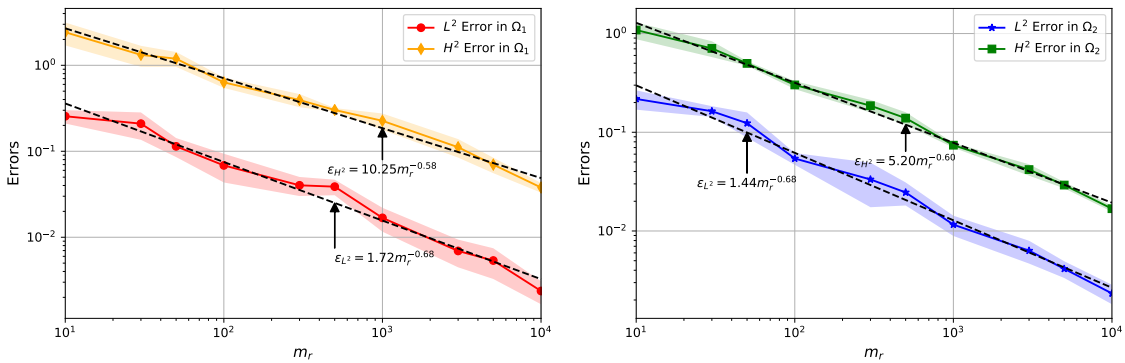

Figure 11: The L^2 and H^2 convergence of the errors of Example 5.5 with respect to the number of training data points. The shaded regions are the one-standard-derivation from five runs with different training data and network initialization. Here, the number of test data points in Ω_1 and Ω_2 are $N_1 = 4044$ and $N_2 = 13787$, respectively.

6 Proofs

We present the proof of Theorem 4.1 in this section. The technique used in the following proof is similar to that used in the proof of Theorem 3 of [29]. However, our Theorem 4.1 applies to elliptic interface problems without a zero-loss assumption of interface and boundary conditions. This prevents direct use of the result from [29], which applies only to elliptic PDEs with a network solution obeying the boundary conditions exactly.

Throughout this section, we assume that Assumptions 4.1 and 4.2 hold. Among the crucial technical tools used here are some Sobolev inequality and probability space filling arguments [43]. We start with the following auxiliary lemma:

Lemma 6.1. *Suppose Assumption 4.1 holds. For training data* $\mathcal{T}_{r_1}^{m_{r_1}} = \{\mathbf{x}_{r_1}^i\}_{i=1}^{m_{r_1}}$, $\mathcal{T}_{r_2}^{m_{r_2}} = \{\mathbf{x}_{r_2}^i\}_{i=1}^{m_{r_2}}$, $\mathcal{T}_b^{m_b} = \{\mathbf{x}_b^i\}_{i=1}^{m_b}$ *and* $\mathcal{T}_{\Gamma}^{m_{\Gamma}} = \{\mathbf{x}_{\Gamma}^i\}_{i=1}^{m_{\Gamma}}$, *if* m_{r_1}, m_{r_2}, m_b *and* m_{Γ} *are large enough to satisfy that there exists* $\mathbf{x}_{r_1}' \in \mathcal{T}_{r_1}$, $\mathbf{x}_{r_2}' \in \mathcal{T}_{r_2}$, $\mathbf{x}_b' \in \mathcal{T}_b$ *and* $\mathbf{x}_{\Gamma}' \in \mathcal{T}_{\Gamma}$ *such that* $\|\mathbf{x}_{r_1} - \mathbf{x}_{r_1}'\|_2 \le \epsilon_{r_1}$, $\|\mathbf{x}_{r_2} - \mathbf{x}_{r_2}'\|_2 \le \epsilon_{r_2}$, $\|\mathbf{x}_b - \mathbf{x}_b'\|_2 \le \epsilon_b$ *and* $\|\mathbf{x}_{\Gamma} - \mathbf{x}_{\Gamma}'\|_2 \le \epsilon_{\Gamma}$ *for any* $\mathbf{x}_{r_1} \in \Omega_1$, $\mathbf{x}_{r_2} \in \Omega_2$, $\mathbf{x}_b \in \partial\Omega$ *and* $\mathbf{x}_{\Gamma} \in \Gamma$, *then, we have*

$$
\begin{aligned}
Loss^{PINN}(u_1,u_2;\boldsymbol{\lambda}) \le C_m \cdot Loss_m^{PINN}(u_1,u_2;\boldsymbol{\lambda}) + 3\lambda_{r_1}\epsilon_{r_1}^2 \left(\left[\mathcal{L}_1[u_1]\right]_{\Omega_1}^2 + \left[f_1\right]_{\Omega_1}^2 \right) \\
+ 3\lambda_{r_2}\epsilon_{r_2}^2 \left(\left[\mathcal{L}_2[u_2]\right]_{\Omega_2}^2 + \left[f_2\right]_{\Omega_2}^2 \right) + 3\lambda_b\epsilon_b^2 \left(\left[\mathcal{B}[u_2]\right]_{\partial\Omega}^2 + \left[g\right]_{\partial\Omega}^2 \right) \\
+ 3\lambda_{\Gamma_D}\epsilon_{\Gamma}^2 \left(\left[\mathcal{I}_D[u_1,u_2]\right]_{\Gamma}^2 + \left[\boldsymbol{\varphi}\right]_{\Gamma}^2 \right) + 3\lambda_{\Gamma_N}\epsilon_{\Gamma}^2 \left(\left[\mathcal{I}_N[u_1,u_2]\right]_{\Gamma}^2 + \left[\boldsymbol{\psi}\right]_{\Gamma}^2 \right),
\end{aligned}
$$

where C_{r_1}, C_{r_2}, C_b, C_{Γ} *are those defined in Assumption 4.1 and*

$$
C_m = 3\max\{C_{r_1}m_{r_1}\epsilon_{r_1}^d, C_{r_2}m_{r_2}\epsilon_{r_2}^d, C_b m_b\epsilon_b^{d-1}, C_{\Gamma}m_{\Gamma}\epsilon_{\Gamma}^{d-1}\}.
$$

Proof. As a consequence of Cauchy's inequality, i.e., $\|\mathbf{x}+\mathbf{y}+\mathbf{z}\|_2^2 \le 3(\|\mathbf{x}\|_2^2+\|\mathbf{y}\|_2^2+\|\mathbf{z}\|_2^2)$ for any three vectors $\mathbf{x},\mathbf{y},\mathbf{z}$, we deduce that for $\mathbf{x}_{r_i}, \mathbf{x}_{r_i}' \in \Omega_i$, $i=1,2$,

$$
\begin{aligned}
&\left|\mathcal{L}_i[u_i](\mathbf{x}_{r_i}) - f_i(\mathbf{x}_{r_i})\right|^2 \\
&\le 3\left(\left|\mathcal{L}_i[u_i](\mathbf{x}_{r_i}) - \mathcal{L}_i[u_i](\mathbf{x}_{r_i}')\right|^2 + \left|\mathcal{L}_i[u_i](\mathbf{x}_{r_i}') - f_i(\mathbf{x}_{r_i}')\right|^2 + \left|f_i(\mathbf{x}_{r_i}') - f_i(\mathbf{x}_{r_i})\right|^2\right).
\end{aligned}
$$

Similarly, for $\mathbf{x}_b, \mathbf{x}_b' \in \partial\Omega$, we have

$$
\begin{aligned}
&\left\|\mathcal{B}[u_2](\mathbf{x}_b) - g(\mathbf{x}_b)\right\|_2^2 \\
&\le 3\left(\left\|\mathcal{B}[u_2](\mathbf{x}_b) - \mathcal{B}[u_2](\mathbf{x}_b')\right\|_2^2 + \left\|\mathcal{B}[u_2](\mathbf{x}_b') - g(\mathbf{x}_b')\right\|_2^2 + \left\|g(\mathbf{x}_b') - g(\mathbf{x}_b)\right\|_2^2\right),
\end{aligned}
$$

and for $\mathbf{x}_{\Gamma}, \mathbf{x}_{\Gamma}' \in \Gamma$, we have

$$
\begin{aligned}
&\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_{\Gamma}) - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma})\right\|_2^2 \\
&\le 3\left(\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_{\Gamma}) - \mathcal{I}_D[u_1,u_2](\mathbf{x}_{\Gamma}')\right\|_2^2 + \left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_{\Gamma}') - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma}')\right\|_2^2 + \left\|\boldsymbol{\varphi}(\mathbf{x}_{\Gamma}) - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma}')\right\|_2^2\right), \\
&\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_{\Gamma}) - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma})\right\|_2^2 \\
&\le 3\left(\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_{\Gamma}) - \mathcal{I}_N[u_1,u_2](\mathbf{x}_{\Gamma}')\right\|_2^2 + \left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_{\Gamma}') - \boldsymbol{\psi}(\mathbf{x}_{\Gamma}')\right\|_2^2 + \left\|\boldsymbol{\psi}(\mathbf{x}_{\Gamma}) - \boldsymbol{\psi}(\mathbf{x}_{\Gamma}')\right\|_2^2\right).
\end{aligned}
$$

In addition, by the conditions, for $\forall \mathbf{x}_{r_1} \in \Omega_1$, $\forall \mathbf{x}_{r_2} \in \Omega_2$, $\forall \mathbf{x}_b \in \partial\Omega$ and $\forall \mathbf{x}_{\Gamma} \in \Gamma$, there exist $\mathbf{x}_{r_1}' \in \mathcal{T}_{r_1}^{m_{r_1}}$, $\mathbf{x}_{r_2}' \in \mathcal{T}_{r_2}^{m_{r_2}}$, $\mathbf{x}_b' \in \mathcal{T}_b^{m_b}$ and $\mathbf{x}_{\Gamma}' \in \mathcal{T}_{\Gamma}^{m_{\Gamma}}$ such that $\|\mathbf{x}_{r_1} - \mathbf{x}_{r_1}'\|_2 \le \epsilon_{r_1}$, $\|\mathbf{x}_{r_2} - \mathbf{x}_{r_2}'\|_2 \le \epsilon_{r_2}$, $\|\mathbf{x}_b - \mathbf{x}_b'\|_2 \le \epsilon_b$ and $\|\mathbf{x}_{\Gamma} - \mathbf{x}_{\Gamma}'\|_2 \le \epsilon_{\Gamma}$. Taking

$$
\begin{aligned}
\mathbf{L}(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_b, \mathbf{x}_{\Gamma}; u_1, u_2, \boldsymbol{\lambda}, \mathbf{0}) = &\lambda_{r_1}\left|\mathcal{L}_1[u_1](\mathbf{x}_{r_1}) - f_1(\mathbf{x}_{r_1})\right|^2 + \lambda_{r_2}\left|\mathcal{L}_2[u_2](\mathbf{x}_{r_2}) - f_2(\mathbf{x}_{r_2})\right|^2 \\
&+ \lambda_b\left\|\mathcal{B}[u_2](\mathbf{x}_b) - g(\mathbf{x}_b)\right\|_2^2 + \lambda_{\Gamma_D}\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_{\Gamma}) - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma})\right\|_2^2 \\
&+ \lambda_{\Gamma_N}\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_{\Gamma}) - \boldsymbol{\psi}(\mathbf{x}_{\Gamma})\right\|_2^2,
\end{aligned}
$$

we have that

$$
\mathbf{L}(\mathbf{x}_{r_1},\mathbf{x}_{r_2},\mathbf{x}_b,\mathbf{x}_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})
$$

$$
\leq 3\mathbf{L}(\mathbf{x}'_{r_1},\mathbf{x}'_{r_2},\mathbf{x}'_b,\mathbf{x}'_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})+3\lambda_{r_1}\left(\left|\mathcal{L}_1[u_1](\mathbf{x}_{r_1})-\mathcal{L}_1[u_1](\mathbf{x}'_{r_1})\right|^2+\left|f_1(\mathbf{x}_{r_1})-f_1(\mathbf{x}'_{r_1})\right|^2\right)
$$

$$
+3\lambda_{r_2}\left(\left|\mathcal{L}_2[u_2](\mathbf{x}_{r_2})-\mathcal{L}_2[u_2](\mathbf{x}'_{r_2})\right|^2+\left|f_2(\mathbf{x}_{r_2})-f_2(\mathbf{x}'_{r_2})\right|^2\right)
$$

$$
+3\lambda_b\left(\left\|\mathcal{B}[u_2](\mathbf{x}_b)-\mathcal{B}[u_2](\mathbf{x}'_b)\right\|_2^2+\left\|\boldsymbol{g}(\mathbf{x}'_b)-\boldsymbol{g}(\mathbf{x}_b)\right\|_2^2\right)
$$

$$
+3\lambda_{\Gamma_D}\left(\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma)-\mathcal{I}_D[u_1,u_2](\mathbf{x}'_\Gamma)\right\|_2^2+\left\|\boldsymbol{\varphi}(\mathbf{x}_\Gamma)-\boldsymbol{\varphi}(\mathbf{x}'_\Gamma)\right\|_2^2\right)
$$

$$
+3\lambda_{\Gamma_N}\left(\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma)-\mathcal{I}_N[u_1,u_2](\mathbf{x}'_\Gamma)\right\|_2^2+\left\|\boldsymbol{\psi}(\mathbf{x}_\Gamma)-\boldsymbol{\psi}(\mathbf{x}'_\Gamma)\right\|_2^2\right)
$$

$$
\leq 3\mathbf{L}(\mathbf{x}'_{r_1},\mathbf{x}'_{r_2},\mathbf{x}'_b,\mathbf{x}'_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})+3\lambda_{r_1}\epsilon_{r_1}^2\left([\mathcal{L}_1[u_1]]_{\Omega_1}^2+[f_1]_{\Omega_1}^2\right)
$$

$$
+3\lambda_{r_2}\epsilon_{r_2}^2\left([\mathcal{L}_2[u_2]]_{\Omega_2}^2+[f_2]_{\Omega_2}^2\right)+3\lambda_b\epsilon_b^2\left([\mathcal{B}[u_2]]_{\partial\Omega}^2+[g]_{\partial\Omega}^2\right)
$$

$$
+3\lambda_{\Gamma_D}\epsilon_\Gamma^2\left([\mathcal{I}_D[u_1,u_2]]_\Gamma^2+[\boldsymbol{\varphi}]_\Gamma^2\right)+3\lambda_{\Gamma_N}\epsilon_\Gamma^2\left([\mathcal{I}_N[u_1,u_2]]_\Gamma^2+[\boldsymbol{\psi}]_\Gamma^2\right).
$$

For $\mathbf{x}_{r_1}^i\in\mathcal{T}_{r_1}^{m_{r_1}}$, $\mathbf{x}_{r_2}^i\in\mathcal{T}_{r_2}^{m_{r_2}}$, $\mathbf{x}_b^i\in\mathcal{T}_b^{m_b}$ and $\mathbf{x}_\Gamma^i\in\mathcal{T}_\Gamma^{m_\Gamma}$, we denote the Voronoi cell associated with $\mathbf{x}_{r_1}^i$, $\mathbf{x}_{r_2}^i$, $\mathbf{x}_b^i$, $\mathbf{x}_\Gamma^i$ as $A_{\mathbf{x}_{r_1}^i}$, $A_{\mathbf{x}_{r_2}^i}$, $A_{\mathbf{x}_b^i}$ and $A_{\mathbf{x}_\Gamma^i}$, respectively, i.e.,

$$
A_{\mathbf{x}_{r_1}^i}=\left\{\mathbf{x}\in\Omega_1\mid\left\|\mathbf{x}-\mathbf{x}_{r_1}^i\right\|_2=\min_{\mathbf{x}'\in\mathcal{T}_{r_1}^{m_{r_1}}}\left\|\mathbf{x}-\mathbf{x}'\right\|_2\right\},
$$

$$
A_{\mathbf{x}_{r_2}^i}=\left\{x\in\Omega_2\mid\left\|\mathbf{x}-\mathbf{x}_{r_2}^i\right\|_2=\min_{\mathbf{x}'\in\mathcal{T}_{r_2}^{m_{r_2}}}\left\|\mathbf{x}-\mathbf{x}'\right\|_2\right\},
$$

$$
A_{\mathbf{x}_b^i}=\left\{\mathbf{x}\in\partial\Omega\mid\left\|x-\mathbf{x}_b^i\right\|_2=\min_{x'\in\mathcal{T}_b^{m_b}}\left\|\mathbf{x}-\mathbf{x}'\right\|_2\right\},
$$

$$
A_{\mathbf{x}_\Gamma^i}=\left\{\mathbf{x}\in\Gamma\mid\left\|\mathbf{x}-\mathbf{x}_\Gamma^i\right\|_2=\min_{x'\in\mathcal{T}_\Gamma^{m_\Gamma}}\left\|\mathbf{x}-\mathbf{x}'\right\|_2\right\},
$$

and let $\omega_{r_1}^i=\mu_{r_1}(A_{\mathbf{x}_{r_1}^i})$, $\omega_{r_2}^i=\mu_{r_2}(A_{\mathbf{x}_{r_2}^i})$, $\omega_b^i=\mu_b(A_{\mathbf{x}_b^i})$ and $\omega_\Gamma^i=\mu_\Gamma(A_{\mathbf{x}_\Gamma^i})$. By taking the expectation with respect to $(\mathbf{x}_{r_1},\mathbf{x}_{r_2},\mathbf{x}_b,\mathbf{x}_\Gamma)\sim\mu=\mu_{r_1}\times\mu_{r_2}\times\mu_b\times\mu_\Gamma$, we obtain that

$$
\mathbb{E}_\mu[\mathbf{L}(\mathbf{x}_{r_1},\mathbf{x}_{r_2},\mathbf{x}_b,\mathbf{x}_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})]
$$

$$
=\sum_{i=1}^{m_\Gamma}\sum_{j=1}^{m_b}\sum_{k=1}^{m_{r_1}}\sum_{l=1}^{m_{r_2}}\int_{A_{\mathbf{x}_\Gamma^i}}\int_{A_{\mathbf{x}_b^j}}\int_{A_{\mathbf{x}_{r_1}^k}}\int_{A_{\mathbf{x}_{r_2}^l}}\mathbf{L}(\mathbf{x}_{r_1},\mathbf{x}_{r_2},\mathbf{x}_b,\mathbf{x}_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})d\mu
$$

$$
\leq 3\sum_{i=1}^{m_\Gamma}\sum_{j=1}^{m_b}\sum_{k=1}^{m_{r_1}}\sum_{l=1}^{m_{r_2}}\omega_\Gamma^i\omega_b^j\omega_{r_1}^k\omega_{r_2}^l\mathbf{L}(\mathbf{x}_{r_1}^k,\mathbf{x}_{r_2}^l,\mathbf{x}_b^j,\mathbf{x}_\Gamma^i;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})+3\lambda_{r_1}\epsilon_{r_1}^2\left([\mathcal{L}_1[u_1]]_{\Omega_1}^2+[f_1]_{\Omega_1}^2\right)
$$

$$
+3\lambda_{r_2}\epsilon_{r_2}^2\left([\mathcal{L}_2[u_2]]_{\Omega_2}^2+[f_2]_{\Omega_2}^2\right)+3\lambda_b\epsilon_b^2\left([\mathcal{B}[u_2]]_{\partial\Omega}^2+[g]_{\partial\Omega}^2\right)
$$

$$
+3\lambda_{\Gamma_D}\epsilon_\Gamma^2\left([\mathcal{I}_D[u_1,u_2]]_\Gamma^2+[\boldsymbol{\varphi}]_\Gamma^2\right)+3\lambda_{\Gamma_N}\epsilon_\Gamma^2\left([\mathcal{I}_N[u_1,u_2]]_\Gamma^2+[\boldsymbol{\psi}]_\Gamma^2\right),
$$

where we have used the fact that $\sum_{i=1}^{m_{r_1}}\omega_{r_1}^i=1$, $\sum_{i=1}^{m_{r_2}}\omega_{r_2}^i=1$, $\sum_{i=1}^{m_b}\omega_b^i=1$ and $\sum_{i=1}^{m_\Gamma}\omega_\Gamma^i=1$.

Next, we give the estimation of the first term on the right. Taking $\omega_{r_1}^{m_{r_1},*}=\max_i\omega_{r_1}^i$, $\omega_{r_2}^{m_{r_2},*}=\max_i\omega_{r_2}^i$, $\omega_b^{m_b,*}=\max_i\omega_b^i$ and $\omega_\Gamma^{m_\Gamma,*}=\max_i\omega_\Gamma^i$, yields that

$$3\sum_{i=1}^{m_\Gamma}\sum_{j=1}^{m_b}\sum_{k=1}^{m_{r_1}}\sum_{l=1}^{m_{r_2}}\omega_\Gamma^i\omega_b^j\omega_{r_1}^k\omega_{r_2}^l\mathbf{L}(\mathbf{x}_{r_1}^k,\mathbf{x}_{r_2}^l,\mathbf{x}_b^j,\mathbf{x}_\Gamma^i;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})$$

$$\leq 3m_{r_1}\omega_{r_1}^{m_{r_1},*}\cdot\frac{\lambda_{r_1}}{m_{r_1}}\sum_{i=1}^{m_{r_1}}\left|\mathcal{L}_1[u_1](\mathbf{x}_{r_1}^i)-f_1(\mathbf{x}_{r_1}^i)\right|^2+3m_{r_2}\omega_{r_2}^{m_{r_2},*}\cdot\frac{\lambda_{r_2}}{m_{r_2}}\sum_{i=1}^{m_{r_2}}\left|\mathcal{L}_2[u_2](\mathbf{x}_{r_2}^i)-f_2(\mathbf{x}_{r_2}^i)\right|^2$$

$$+3m_\Gamma\omega_\Gamma^{m_\Gamma,*}\cdot\left(\frac{\lambda_{\Gamma_D}}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma^i)-\boldsymbol{\varphi}(\mathbf{x}_\Gamma^i)\right\|_2^2+\frac{\lambda_{\Gamma_N}}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma^i)-\boldsymbol{\psi}(\mathbf{x}_\Gamma^i)\right\|_2^2\right)$$

$$+3m_b\omega_b^{m_b,*}\cdot\frac{\lambda_b}{m_b}\sum_{i=1}^{m_b}\left\|\mathcal{B}[u_2](\mathbf{x}_b^i)-\mathbf{g}(\mathbf{x}_b^i)\right\|_2^2, \tag{6.1}$$

where we have used the fact that $m_{r_1}\omega_{r_1}^{m_{r_1},*},m_{r_2}\omega_{r_2}^{m_{r_2},*},m_b\omega_b^{m_b,*},m_\Gamma\omega_\Gamma^{m_\Gamma,*}\geq 1$.

Let $B_\epsilon(\mathbf{x})$ be a closed ball centered at $\mathbf{x}$ with radius ϵ. Let $P_{r_1}^*=\max_{\mathbf{x}\in\Omega_1}\mu_{r_1}(B_{\epsilon_{r_1}}(\mathbf{x})\cap\Omega_1)$, $P_{r_2}^*=\max_{\mathbf{x}\in\Omega_2}\mu_{r_2}(B_{\epsilon_{r_2}}(\mathbf{x})\cap\Omega_2)$, $P_b^*=\max_{\mathbf{x}\in\partial\Omega}\mu_b(B_{\epsilon_b}(\mathbf{x})\cap\partial\Omega)$ and $P_\Gamma^*=\max_{\mathbf{x}\in\Gamma}\mu_\Gamma(B_{\epsilon_\Gamma}(\mathbf{x})\cap\Gamma)$. Then for any $\mathbf{x}_{r_1}\in\Omega_1$, $\mathbf{x}_{r_2}\in\Omega_2$, $\mathbf{x}_b\in\partial\Omega$ and $\mathbf{x}_\Gamma\in\Gamma$, there exists $\mathbf{x}_{r_1}'\in\mathcal{T}_{r_1}$, $\mathbf{x}_{r_2}'\in\mathcal{T}_{r_2}$, $\mathbf{x}_b'\in\mathcal{T}_b$ and $\mathbf{x}_\Gamma'\in\mathcal{T}_\Gamma$ such that $\left\|\mathbf{x}_{r_1}-\mathbf{x}_{r_1}'\right\|_2\leq\epsilon_{r_1}$, $\left\|\mathbf{x}_{r_2}-\mathbf{x}_{r_2}'\right\|_2\leq\epsilon_{r_2}$, $\left\|\mathbf{x}_b-\mathbf{x}_b'\right\|_2\leq\epsilon_b$ and $\left\|\mathbf{x}_\Gamma-\mathbf{x}_\Gamma'\right\|_2\leq\epsilon_\Gamma$ for each i, there are closed balls $B_{\epsilon_{r_1}}$, $B_{\epsilon_{r_2}}$, B_{ϵ_b} and B_{ϵ_Γ} that include $A_{\mathbf{x}_{r_1}^i}$, $A_{\mathbf{x}_{r_2}^i}$, $A_{\mathbf{x}_b^i}$ and $A_{\mathbf{x}_\Gamma^i}$, respectively. These facts, together with Assumption 4.1 imply that

$$\omega_{r_1}^{m_{r_1},*}\leq P_{r_1}^*\leq C_{r_1}\epsilon_{r_1}^d,\quad \omega_{r_2}^{m_{r_2},*}\leq P_{r_2}^*\leq C_{r_2}\epsilon_{r_2}^d,\quad \omega_b^{m_b,*}\leq P_b^*\leq C_b\epsilon_b^{d-1},\quad \omega_\Gamma^{m_\Gamma,*}\leq P_\Gamma^*\leq C_\Gamma\epsilon_\Gamma^{d-1}. \tag{6.2}$$

With the estimations (6.1) and (6.2), we obtain that

$$\mathbb{E}_\mu\left[\mathbf{L}(\mathbf{x}_{r_1},\mathbf{x}_{r_2},\mathbf{x}_b,\mathbf{x}_\Gamma;u_1,u_2,\boldsymbol{\lambda},\mathbf{0})\right]$$

$$\leq 3C_{r_1}m_{r_1}\epsilon_{r_1}^d\cdot\frac{\lambda_{r_1}}{m_{r_1}}\sum_{i=1}^{m_{r_1}}\left|\mathcal{L}_1[u_1](\mathbf{x}_{r_1}^i)-f_1(\mathbf{x}_{r_1}^i)\right|^2+3C_{r_2}m_{r_2}\epsilon_{r_2}^d\cdot\frac{\lambda_{r_2}}{m_{r_2}}\sum_{i=1}^{m_{r_2}}\left|\mathcal{L}_2[u_2](\mathbf{x}_{r_2}^i)-f_2(\mathbf{x}_{r_2}^i)\right|^2$$

$$+3C_\Gamma m_\Gamma\epsilon_\Gamma^{d-1}\cdot\left(\frac{\lambda_{\Gamma_D}}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_D[u_1,u_2](\mathbf{x}_\Gamma^i)-\boldsymbol{\varphi}(\mathbf{x}_\Gamma^i)\right\|_2^2+\frac{\lambda_{\Gamma_N}}{m_\Gamma}\sum_{i=1}^{m_\Gamma}\left\|\mathcal{I}_N[u_1,u_2](\mathbf{x}_\Gamma^i)-\boldsymbol{\psi}(\mathbf{x}_\Gamma^i)\right\|_2^2\right)$$

$$+3C_b m_b\epsilon_b^{d-1}\cdot\frac{\lambda_b}{m_b}\sum_{j=1}^{m_b}\left\|\mathcal{B}[u_2](\mathbf{x}_b^j)-\mathbf{g}(\mathbf{x}_b^j)\right\|_2^2+3\lambda_{r_1}\epsilon_{r_1}^2\left([\mathcal{L}_1[u_1]]_{\Omega_1}^2+[f_1]_{\Omega_1}^2\right)$$

$$+3\lambda_{r_2}\epsilon_{r_2}^2\left([\mathcal{L}_2[u_2]]_{\Omega_2}^2+[f_2]_{\Omega_2}^2\right)+3\lambda_b\epsilon_b^2\left([\mathcal{B}[u_2]]_{\partial\Omega}^2+[\mathbf{g}]_{\partial\Omega}^2\right)$$

$$+3\lambda_{\Gamma_D}\epsilon_\Gamma^2\left([\mathcal{I}_D[u_1,u_2]]_\Gamma^2+[\boldsymbol{\varphi}]_\Gamma^2\right)+3\lambda_{\Gamma_N}\epsilon_\Gamma^2\left([\mathcal{I}_N[u_1,u_2]]_\Gamma^2+[\boldsymbol{\psi}]_\Gamma^2\right).$$

Finally, we conclude the proof by taking

$$C_{\boldsymbol{m}}=3\max\{C_{r_1}m_{r_1}\epsilon_{r_1}^d,C_{r_2}m_{r_2}\epsilon_{r_2}^d,C_b m_b\epsilon_b^{d-1},C_\Gamma m_\Gamma\epsilon_\Gamma^{d-1}\}. \qquad\square$$

With Lemma 6.1 and Assumption 4.1, we are able to quantify the generalization error and provide an upper bound of the expected unregularized PINN loss (3.3).

Lemma 6.2. *Suppose Assumption 4.1 holds. Suppose that u_1, u_2 satisfy*

$$R_{r_1}(u_1) < \infty, \quad R_{r_2}(u_2) < \infty, \quad R_b(u_2) < \infty, \quad R_{\Gamma_D}(u_1, u_2) < \infty, \quad R_{\Gamma_N}(u_1, u_2) < \infty,$$

and $f_1, f_2, \psi, \varphi, g$ satisfy

$$[f_1]_{\Omega_1}, \ [f_2]_{\Omega_2}, \ [g]_{\partial\Omega}, \ [\varphi]_\Gamma, \ [\psi]_\Gamma < \infty.$$

Let m_{r_1}, m_{r_2}, m_b and m_Γ be the number of iid samples from μ_{r_1}, μ_{r_2}, μ_b and μ_Γ, respectively. Let $\boldsymbol{\lambda} = (\lambda_{r_1}, \lambda_{r_2}, \lambda_b, \lambda_{\Gamma_D}, \lambda_{\Gamma_N})$ be a fixed vector. Then, with probability at least,

$$P = P(m_{r_1})P(m_{r_2})P(m_b)P(m_\Gamma), \quad \text{where} \quad P(m) = (1 - \sqrt{m}(1 - 1/\sqrt{m})^m),$$

we have

$$Loss^{PINN}(u_1, u_2; \boldsymbol{\lambda}) \leq C_m \cdot Loss_m(u_1, u_2; \boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}_m^R) + C'(m_{r_1}^{-\frac{1}{d}} + m_{r_2}^{-\frac{1}{d}} + m_b^{-\frac{1}{d-1}} + m_\Gamma^{-\frac{1}{d-1}}).$$

Here, $\hat{\boldsymbol{\lambda}}_m^R = (\hat{\lambda}_{r_1,m}^R, \hat{\lambda}_{r_2,m}^R, \hat{\lambda}_{b,m}^R, \hat{\lambda}_{\Gamma_D,m}^R, \hat{\lambda}_{\Gamma_N,m}^R)$. Specifically,

$$\hat{\lambda}_{r_1,m}^R = \frac{3\lambda_{r_1} d c_{r_1}^{-\frac{2}{d}}}{C_m} \cdot m_{r_1}^{-\frac{1}{d}}, \quad \hat{\lambda}_{r_2,m}^R = \frac{3\lambda_{r_2} d c_{r_2}^{-\frac{2}{d}}}{C_m} \cdot m_{r_2}^{-\frac{1}{d}}, \quad \hat{\lambda}_{b,m}^R = \frac{3\lambda_b d c_b^{-\frac{2}{d-1}}}{C_m} \cdot m_b^{-\frac{1}{d-1}},$$

$$\hat{\lambda}_{\Gamma_D,m}^R = \frac{3\lambda_{\Gamma_D} d c_\Gamma^{-\frac{2}{d-1}}}{C_m} \cdot m_\Gamma^{-\frac{1}{d-1}}, \quad \hat{\lambda}_{\Gamma_N,m}^R = \frac{3\lambda_{\Gamma_N} d c_\Gamma^{-\frac{2}{d-1}}}{C_m} \cdot m_\Gamma^{-\frac{1}{d-1}}.$$

$C_m = 3\max\{\kappa_{r_1}\sqrt{d}^d m_{r_1}^{\frac{1}{2}}, \kappa_{r_2}\sqrt{d}^d m_{r_2}^{\frac{1}{2}}, \kappa_b\sqrt{d}^{d-1} m_b^{\frac{1}{2}}, \kappa_\Gamma\sqrt{d}^{d-1} m_\Gamma^{\frac{1}{2}}\}$ where $\kappa_{r_1} = \frac{C_{r_1}}{c_{r_1}}$, $\kappa_{r_2} = \frac{C_{r_2}}{c_{r_2}}$, $\kappa_b = \frac{C_b}{c_b}$, $\kappa_\Gamma = \frac{C_\Gamma}{c_\Gamma}$. And C' is a constant that depends only on $\boldsymbol{\lambda}, d, c_{r_1}, c_{r_2}, c_b, c_\Gamma, f_1, f_2, g, \varphi, \psi$.

Proof. Since $\mathcal{T}_{r_1} = \{\mathbf{x}_{r_1}^i\}_{i=1}^{m_{r_1}}$ be iid samples from μ_{r_1} on Ω_1, $\mathcal{T}_{r_2} = \{\mathbf{x}_{r_2}^i\}_{i=1}^{m_{r_2}}$ be iid samples from μ_{r_2} on Ω_2, $\mathcal{T}_b = \{\mathbf{x}_b^i\}_{i=1}^{m_b}$ be iid samples from μ_b on $\partial\Omega$ and $\mathcal{T}_\Gamma = \{\mathbf{x}_\Gamma^i\}_{i=1}^{m_\Gamma}$ be iid samples from μ_Γ on Γ, respectively, therefore, by Lemma B.2 in [29], with probability at least

$$P = P(m_{r_1})P(m_{r_2})P(m_b)P(m_\Gamma), \quad \text{where} \quad P(m) = (1 - \sqrt{m}(1 - 1/\sqrt{m})^m), \quad (6.3)$$

for $\forall \mathbf{x}_{r_1} \in \Omega_1$, $\forall \mathbf{x}_{r_2} \in \Omega_2$, $\forall \mathbf{x}_b \in \partial\Omega$ and $\forall \mathbf{x}_\Gamma \in \Gamma$, there exists $\mathbf{x}'_{r_1} \in \mathcal{T}_{r_1}^{m_{r_1}}$, $\mathbf{x}'_{r_2} \in \mathcal{T}_{r_2}^{m_{r_2}}$, $\mathbf{x}'_b \in \mathcal{T}_b^{m_b}$ and $\mathbf{x}'_\Gamma \in \mathcal{T}_\Gamma^{m_\Gamma}$ such that $\|\mathbf{x}_{r_1} - \mathbf{x}'_{r_1}\|_2 \leq \sqrt{d}c_{r_1}^{-\frac{1}{d}} m_{r_1}^{-\frac{1}{2d}}$, $\|\mathbf{x}_{r_2} - \mathbf{x}'_{r_2}\|_2 \leq \sqrt{d}c_{r_2}^{-\frac{1}{d}} m_{r_2}^{-\frac{1}{2d}}$, $\|\mathbf{x}_b - \mathbf{x}'_b\|_2 \leq \sqrt{d}c_b^{-\frac{1}{d-1}} m_b^{-\frac{1}{2(d-1)}}$ and $\|\mathbf{x}_\Gamma - \mathbf{x}'_\Gamma\|_2 \leq \sqrt{d}c_\Gamma^{-\frac{1}{d-1}} m_\Gamma^{-\frac{1}{2(d-1)}}$.

Using Lemma 6.1, together with taking $\epsilon_{r_1} = \sqrt{d}c_{r_1}^{-\frac{1}{d}}m_{r_1}^{-\frac{1}{2d}}$, $\epsilon_{r_2} = \sqrt{d}c_{r_2}^{-\frac{1}{d}}m_{r_2}^{-\frac{1}{2d}}$, $\epsilon_b = \sqrt{d}c_b^{-\frac{1}{d-1}}m_b^{-\frac{1}{2(d-1)}}$ and $\epsilon_\Gamma = \sqrt{d}c_\Gamma^{-\frac{1}{d-1}}m_\Gamma^{-\frac{1}{2(d-1)}}$, implies that with probability at least (6.3),

$$
\begin{aligned}
\text{Loss}^{\text{PINN}}(u_1,u_2;\lambda) \leq\ & C_m \cdot \text{Loss}_m^{\text{PINN}}(u_1,u_2;\lambda) + Q \\
& + C_m \cdot \left[\hat{\lambda}_{r_1,m}^R \cdot [\mathcal{L}_1[u_1]]_{\Omega_1}^2 + \hat{\lambda}_{r_2,m}^R \cdot [\mathcal{L}_2[u_2]]_{\Omega_2}^2 + \hat{\lambda}_{b,m}^R \cdot [\mathcal{B}[u_2]]_{\partial\Omega}^2 \right] \\
& + C_m \cdot \left[\hat{\lambda}_{\Gamma_D,m}^R \cdot [\mathcal{I}_D[u_1,u_2]]_\Gamma^2 + \hat{\lambda}_{\Gamma_N,m}^R \cdot [\mathcal{I}_N[u_1,u_2]]_\Gamma^2 \right],
\end{aligned}
$$

where

$$
\begin{aligned}
Q = \ & 3\lambda_{r_1}dc_{r_1}^{-\frac{2}{d}}m_{r_1}^{-\frac{1}{d}}[f_1]_{\Omega_1}^2 + 3\lambda_{r_2}dc_{r_2}^{-\frac{2}{d}}m_{r_2}^{-\frac{1}{d}}[f_2]_{\Omega_2}^2 + 3\lambda_b dc_b^{-\frac{2}{d-1}}m_b^{-\frac{1}{d-1}}[g]_{\partial\Omega}^2 \\
& + 3dc_\Gamma^{-\frac{2}{d-1}}m_\Gamma^{-\frac{1}{d-1}}\left(\lambda_{\Gamma_D}[\boldsymbol{\varphi}]_\Gamma^2 + \lambda_{\Gamma_N}[\boldsymbol{\psi}]_\Gamma^2 \right),
\end{aligned}
$$

and

$$
C_m = 3\max\left\{ \frac{C_{r_1}}{c_{r_1}}\sqrt{d}^{\,d}m_{r_1}^{\frac{1}{2}}, \frac{C_{r_2}}{c_{r_2}}\sqrt{d}^{\,d}m_{r_2}^{\frac{1}{2}}, \frac{C_b}{c_b}\sqrt{d}^{\,d-1}m_b^{\frac{1}{2}}, \frac{C_\Gamma}{c_\Gamma}\sqrt{d}^{\,d-1}m_\Gamma^{\frac{1}{2}} \right\},
$$

$$
\hat{\lambda}_{r_1,m}^R = \frac{3\lambda_{r_1}dc_{r_1}^{-\frac{2}{d}}}{C_m}\cdot m_{r_1}^{-\frac{1}{d}}, \quad \hat{\lambda}_{r_2,m}^R = \frac{3\lambda_{r_2}dc_{r_2}^{-\frac{2}{d}}}{C_m}\cdot m_{r_2}^{-\frac{1}{d}}, \quad \hat{\lambda}_{b,m}^R = \frac{3\lambda_b dc_b^{-\frac{2}{d-1}}}{C_m}\cdot m_b^{-\frac{1}{d-1}},
$$

$$
\hat{\lambda}_{\Gamma_D,m}^R = \frac{3\lambda_{\Gamma_D}dc_\Gamma^{-\frac{2}{d-1}}}{C_m}\cdot m_\Gamma^{-\frac{1}{d-1}}, \quad \hat{\lambda}_{\Gamma_N,m}^R = \frac{3\lambda_{\Gamma_N}dc_\Gamma^{-\frac{2}{d-1}}}{C_m}\cdot m_\Gamma^{-\frac{1}{d-1}}.
$$

By taking

$$
C' = 3\max\left\{ \lambda_{r_1}dc_{r_1}^{-\frac{2}{d}}[f_1]_{\Omega_1}^2, \lambda_{r_2}dc_{r_2}^{-\frac{2}{d}}[f_2]_{\Omega_2}^2, \lambda_b dc_b^{-\frac{2}{d-1}}[g]_{\partial\Omega}^2, dc_\Gamma^{-\frac{2}{d-1}}\left(\lambda_{\Gamma_D}[\boldsymbol{\varphi}]_\Gamma^2 + \lambda_{\Gamma_N}[\boldsymbol{\psi}]_\Gamma^2 \right) \right\},
$$

we conclude that

$$
\text{Loss}^{\text{PINN}}(u_1,u_2;\lambda) \leq C_m \cdot \text{Loss}_m(u_1,u_2;\lambda,\hat{\lambda}_m^R) + C'(m_{r_1}^{-\frac{1}{d}} + m_{r_2}^{-\frac{1}{d}} + m_b^{-\frac{1}{d-1}} + m_\Gamma^{-\frac{1}{d-1}}),
$$

where $\hat{\lambda}_m^R = (\hat{\lambda}_{r_1,m}^R, \hat{\lambda}_{r_2,m}^R, \hat{\lambda}_{b,m}^R, \hat{\lambda}_{\Gamma_D,m}^R, \hat{\lambda}_{\Gamma_N,m}^R)$. The proof is completed. $\square$

Using Lemma 6.2, we will show that the expected PINN loss (3.3) at the minimizers of the Lipschitz regularized empirical loss (3.2) converges to zero according to Assumptions 4.2.

Lemma 6.3. *Suppose Assumptions 4.1 and 4.2 hold. Let m_{r_1}, m_{r_2}, m_Γ and m_b be the number of iid samples from μ_{r_1}, μ_{r_2}, μ_Γ and μ_b, respectively, and satisfy $m_{r_2} = \mathcal{O}(m_{r_1})$, $m_\Gamma = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$, $m_b = \mathcal{O}(m_{r_1}^{\frac{d-1}{d}})$. Let λ_m^R be a vector satisfying*

$$
\lambda_m^R \geq \hat{\lambda}_m^R, \quad \left\| \lambda_m^R \right\|_\infty = \mathcal{O}(\left\| \hat{\lambda}_m^R \right\|_\infty),
$$

where $\hat{\boldsymbol{\lambda}}_m^R = (\hat{\lambda}_{r_1,m}^R, \; \hat{\lambda}_{r_2,m}^R, \; \hat{\lambda}_{b,m}^R, \; \hat{\lambda}_{\Gamma_D,m}^R, \; \hat{\lambda}_{\Gamma_N,m}^R)$ *are defined in Lemma 6.2. Let* $(u_{1,m},u_{2,m}) \in$
$(\mathcal{H}_{1,m},\mathcal{H}_{2,m})$ *be a minimizer of the Lipschitz regularized empirical loss* $Loss_m(\cdot;\boldsymbol{\lambda},\boldsymbol{\lambda}_m^R)$ *(3.2).*
Then the following holds:

- *With probability at least*

$$P = P(m_{r_1})P(m_{r_2})P(m_b)P(m_\Gamma), \quad where \quad P(m) = (1 - \sqrt{m}(1 - 1/\sqrt{m})^m)$$

  over iid samples,

$$Loss^{PINN}(u_{1,m},u_{2,m};\boldsymbol{\lambda}) = \mathcal{O}(m_{r_1}^{-\frac{1}{d}}).$$

- *With probability 1 over iid samples,*

$$\lim_{m_{r_1}\to\infty}\mathcal{L}[u_{1,m}] = f_1 \text{ in } L^2(\Omega_1), \quad \lim_{m_{r_1}\to\infty}\mathcal{L}[u_{2,m}] = f_2 \text{ in } L^2(\Omega_2),$$

$$\lim_{m_{r_1}\to\infty}\mathcal{B}[u_{2,m}] = g \text{ in } L^2(\partial\Omega), \quad \lim_{m_{r_1}\to\infty}\mathcal{I}_D[u_{1,m},u_{2,m}] = \boldsymbol{\varphi} \text{ in } L^2(\Gamma),$$

$$\lim_{m_{r_1}\to\infty}\mathcal{I}_N[u_{1,m},u_{2,m}] = \boldsymbol{\psi} \text{ in } L^2(\Gamma).$$

Proof. Since $m_{r_1} = \mathcal{O}(m_{r_2}) = \mathcal{O}(m_b^{\frac{d}{d-1}}) = \mathcal{O}(m_\Gamma^{\frac{d}{d-1}})$, we have

$$\hat{\lambda}_{r_1,m}^R, \; \hat{\lambda}_{r_2,m}^R, \; \hat{\lambda}_{b,m}^R, \; \hat{\lambda}_{\Gamma_D,m}^R, \; \hat{\lambda}_{\Gamma_N,m}^R = \mathcal{O}(m_{r_1}^{-\frac{1}{2}-\frac{1}{d}}),$$

where $\hat{\lambda}_{r_1,m}^R, \; \hat{\lambda}_{r_2,m}^R, \; \hat{\lambda}_{b,m}^R, \; \hat{\lambda}_{\Gamma_D,m}^R, \; \hat{\lambda}_{\Gamma_N,m}^R$ are defined in Lemma 6.2. Let $\boldsymbol{\lambda}$ be a vector independent of $\boldsymbol{m}$ and $\boldsymbol{\lambda}_m^R = (\lambda_{r_1,m}^R, \; \lambda_{r_2,m}^R, \; \lambda_{b,m}^R, \; \lambda_{\Gamma_D,m}^R, \; \lambda_{\Gamma_N,m}^R)$ be a vector satisfying

$$\boldsymbol{\lambda}_m^R \geq \hat{\boldsymbol{\lambda}}_m^R, \qquad \left\|\boldsymbol{\lambda}_m^R\right\|_\infty = \mathcal{O}\left(\left\|\hat{\boldsymbol{\lambda}}_m^R\right\|_\infty\right),$$

where $\hat{\boldsymbol{\lambda}}_m^R = (\hat{\lambda}_{r_1,m}^R, \; \hat{\lambda}_{r_2,m}^R, \; \hat{\lambda}_{b,m}^R, \; \hat{\lambda}_{\Gamma_D,m}^R, \; \hat{\lambda}_{\Gamma_N,m}^R)$. Let $(u_{1,m},u_{2,m}) \in (\mathcal{H}_{1,m},\mathcal{H}_{2,m})$ minimizes the Lipschitz regularized loss $Loss_m(\cdot;\boldsymbol{\lambda},\boldsymbol{\lambda}_m^R)$ (3.2). Let $(\hat{u}_{1,m},\hat{u}_{2,m})$ be the neural networks defined in the third term of Assumption 4.2, i.e., they satisfy $Loss_m^{PINN}(\hat{u}_{1,m},\hat{u}_{2,m};\boldsymbol{\lambda}) = \mathcal{O}(m_{r_1}^{-\frac{1}{2}-\frac{1}{d}})$.

Then, we have

$$
\begin{aligned}
Loss_m(u_{1,m},u_{2,m};\boldsymbol{\lambda},\boldsymbol{\lambda}_m^R) \leq & Loss_m(\hat{u}_{1,m},\hat{u}_{2,m};\boldsymbol{\lambda},\boldsymbol{\lambda}_m^R) \\
\leq & \left\|\boldsymbol{\lambda}_m^R\right\|_\infty \big(R_{r_1}(\hat{u}_{1,m}) + R_{r_2}(\hat{u}_{2,m}) + R_{\Gamma_D}(\hat{u}_{2,m},\hat{u}_{1,m}) + R_{\Gamma_N}(\hat{u}_{2,m},\hat{u}_{1,m}) \\
& + R_b(\hat{u}_{2,m})\big) + Loss_m(\hat{u}_{1,m},\hat{u}_{2,m};\boldsymbol{\lambda},0).
\end{aligned}
$$

Let

$$\hat{R} = \sup_m \big(R_{r_1}(\hat{u}_{1,m}) + R_{r_2}(\hat{u}_{2,m}) + R_{\Gamma_D}(\hat{u}_{1,m},\hat{u}_{2,m}) + R_{\Gamma_N}(\hat{u}_{1,m},\hat{u}_{2,m}) + R_b(\hat{u}_{2,m})\big).$$

Note that $\text{Loss}_m^{\text{PINN}}(\hat{u}_{1,m}, \hat{u}_{2,m}; \boldsymbol{\lambda}) = \text{Loss}_m(\hat{u}_{1,m}, \hat{u}_{2,m}; \boldsymbol{\lambda}, 0)$ and $\left\| \boldsymbol{\lambda}_m^R \right\|_\infty = \mathcal{O}\left(\left\| \hat{\boldsymbol{\lambda}}_m^R \right\|_\infty \right) = \mathcal{O}\left(m_{r_1}^{-\frac{1}{2} - \frac{1}{d}} \right)$. By the last term in Assumption 4.2, we have $\hat{R} < \infty$. Therefore,

$$\text{Loss}_m(u_{1,m}, u_{2,m}; \boldsymbol{\lambda}, \boldsymbol{\lambda}_m^R) = \mathcal{O}\left(m_{r_1}^{-\frac{1}{2} - \frac{1}{d}} \right).$$

According to Lemma 6.2, with probability at least

$$P = P(m_{r_1})P(m_{r_2})P(m_b)P(m_\Gamma), \quad \text{where} \quad P(m) = (1 - \sqrt{m}(1 - 1/\sqrt{m})^m),$$

we have that

$$\text{Loss}^{\text{PINN}}(u_{1,m}, u_{2,m}; \boldsymbol{\lambda})$$
$$\leq C_m \cdot \text{Loss}_m(u_{1,m}, u_{2,m}; \boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}_m^R) + C'\left(m_{r_1}^{-\frac{1}{d}} + m_{r_2}^{-\frac{1}{d}} + m_b^{-\frac{1}{d-1}} + m_\Gamma^{-\frac{1}{d-1}} \right) = \mathcal{O}\left(m_{r_1}^{-\frac{1}{d}} \right),$$

which completes the first part of the proof. Here, we have used the fact that $m_{r_2} = \mathcal{O}(m_{r_1})$, $m_\Gamma = \mathcal{O}\left(m_{r_1}^{\frac{d-1}{d}} \right)$, $m_b = \mathcal{O}\left(m_{r_1}^{\frac{d-1}{d}} \right)$ and $C_m = \mathcal{O}\left(m_{r_1}^{\frac{1}{2}} \right)$.

In addition, by the first part of the Lemma, we have that the probability of

$$\lim_{m_{r_1} \to \infty} \text{Loss}(u_{1,m}, u_{2,m}; \boldsymbol{\lambda}, \mathbf{0}) = 0$$

is one. Consequently, with probability one over iid samples,

$$0 = \lim_{m_{r_1} \to \infty} \text{Loss}(u_{1,m}, u_{2,m}; \boldsymbol{\lambda}, \mathbf{0})$$
$$= \lim_{m_{r_1} \to \infty} \left(\lambda_{r_1} \int_{\Omega_1} |\mathcal{L}_1[u_{1,m}](\mathbf{x}_{r_1}) - f_1(\mathbf{x}_{r_1})|^2 d\mu_{r_1}(\mathbf{x}_{r_1}) \right.$$
$$+ \lambda_{r_2} \int_{\Omega_2} |\mathcal{L}_2[u_{2,m}](\mathbf{x}_{r_2}) - f_2(\mathbf{x}_{r_2})|^2 d\mu_{r_2}(\mathbf{x}_{r_2})$$
$$+ \lambda_b \int_{\partial\Omega} \|\mathcal{B}[u_{2,m}](\mathbf{x}_b) - \mathbf{g}(\mathbf{x}_b)\|_2^2 d\mu_b(\mathbf{x}_b)$$
$$+ \lambda_{\Gamma_D} \int_\Gamma \|\mathcal{I}_D[u_{1,m}, u_{2,m}](\mathbf{x}_\Gamma) - \boldsymbol{\varphi}(\mathbf{x}_\Gamma)\|_2^2 d\mu_\Gamma(\mathbf{x}_\Gamma)$$
$$\left. + \lambda_{\Gamma_N} \int_\Gamma \|\mathcal{I}_N[u_{1,m}, u_{2,m}](\mathbf{x}_\Gamma) - \boldsymbol{\psi}(\mathbf{x}_\Gamma)\|_2^2 d\mu_\Gamma(\mathbf{x}_\Gamma) \right).$$

Since $\boldsymbol{\lambda} = (\lambda_{r_1}, \lambda_{r_2}, \lambda_b, \lambda_{\Gamma_D}, \lambda_{\Gamma_N}) \geq 0$, we obtain that

$$\lim_{m_{r_1} \to \infty} \int_{\Omega_1} |\mathcal{L}_1[u_{1,m}](\mathbf{x}_{r_1}) - f_1(\mathbf{x}_{r_1})|^2 d\mu_{r_1}(\mathbf{x}_{r_1}) = 0,$$

$$\lim_{m_{r_1} \to \infty} \int_{\Omega_2} |\mathcal{L}_2[u_{2,m}](\mathbf{x}_{r_2}) - f_2(\mathbf{x}_{r_2})|^2 d\mu_{r_2}(\mathbf{x}_{r_2}) = 0,$$

$$\lim_{m_{r_1} \to \infty} \int_{\partial\Omega} \|\mathcal{B}[u_{2,m}](\mathbf{x}_b) - \boldsymbol{g}(\mathbf{x}_b)\|_2^2 d\mu_b(\mathbf{x}_b) = 0,$$

$$\lim_{m_{r_1} \to \infty} \int_{\Gamma} \|\mathcal{I}_D[u_{1,m}, u_{2,m}](\mathbf{x}_{\Gamma}) - \boldsymbol{\varphi}(\mathbf{x}_{\Gamma})\|_2^2 d\mu_{\Gamma}(\mathbf{x}_{\Gamma}) = 0,$$

$$\lim_{m_{r_1} \to \infty} \int_{\Gamma} \|\mathcal{I}_N[u_{1,m}, u_{2,m}](\mathbf{x}_{\Gamma}) - \boldsymbol{\psi}(\mathbf{x}_{\Gamma})\|_2^2 d\mu_{\Gamma}(\mathbf{x}_{\Gamma}) = 0.$$

Therefore, we conclude that $\mathcal{L}_1[u_{1,m}] \to f_1$ in $L^2(\Omega_1; \mu_{r_1})$, $\mathcal{L}_2[u_{2,m}] \to f_2$ in $L^2(\Omega_2; \mu_{r_2})$, $\mathcal{B}[u_{2,m}] \to \boldsymbol{g}$ in $L^2(\partial\Omega; \mu_b)$, $\mathcal{I}_D[u_{1,m}, u_{2,m}] \to \boldsymbol{\varphi}$ in $L^2(\Gamma; \mu_{\Gamma})$ and $\mathcal{I}_N[u_{1,m}, u_{2,m}] \to \boldsymbol{\psi}$ in $L^2(\Gamma; \mu_{\Gamma})$ as $m_{r_1} \to \infty$. □

Finally, to complete the proof, it is sufficient to present the following estimate for the interface problem (2.4). For convenience, we denote $X = H^2(\Omega_1) \cap H^2(\Omega_2)$ and define

$$\|u\|_X = \|u\|_{H^2(\Omega_1)} + \|u\|_{H^2(\Omega_2)}, \quad \forall u \in X.$$

Lemma 6.4. *Assume that* $\varphi \in H^2(\Gamma)$, $\psi \in H^1(\Gamma)$, $g \in H^2(\partial\Omega)$, $f_i \in L^2(\Omega)$, $i = 1, 2$. *Then the problem* (2.4) *has a unique solution* $u \in X$ *and* u *satisfies the estimate:*

$$\|u\|_X \le C \left(\|f\|_{L^2(\Omega)} + \|g\|_{H^2(\partial\Omega)} + \|\varphi\|_{H^2(\Gamma)} + \|\psi\|_{H^1(\Gamma)} \right).$$

Proof. Let $\tilde{u}_1$ solve

$$-\Delta\tilde{u}_1 = 0, \quad \text{in } \Omega_1,$$
$$\tilde{u}_1 = -\varphi \quad \text{on } \Gamma.$$

We know $\tilde{u}_1$ exists and $\tilde{u}_1 \in H^2(\Omega_1)$ satisfying (cf. Grisvard [51])

$$\|\tilde{u}_1\|_{H^2(\Omega_1)} \le c \|\varphi\|_{H^{3/2}(\Gamma)}.$$

Let $\tilde{u}_2$ solve

$$-\Delta^2\tilde{u}_2 = 0, \quad \text{in } \Omega_2,$$
$$\tilde{u}_2 = 0, \quad \frac{\partial\tilde{u}_2}{\partial n} = \frac{a_1}{a_2}\frac{\partial\tilde{u}_1}{\partial n} + \frac{\psi}{a_2} \quad \text{on } \Gamma,$$
$$\tilde{u}_2 = g, \quad \frac{\partial\tilde{u}_2}{\partial\nu} = 0 \quad \text{on } \partial\Omega.$$

We know $\tilde{u}_2$ exists and $\tilde{u}_2 \in H^2(\Omega_2)$ satisfying (cf. Girault-Raviart [52], pp.15-17)

$$\|\tilde{u}_2\|_{H^2(\Omega_2)} \le C \left(\|g\|_{H^{3/2}(\partial\Omega)} + \left\|\frac{\partial\tilde{u}_1}{\partial n}\right\|_{H^{1/2}(\Gamma)} + \|\psi\|_{H^{1/2}(\Gamma)} \right)$$
$$\le C \left(\|g\|_{H^{3/2}(\partial\Omega)} + \|\tilde{u}_1\|_{H^2(\Omega_1)} + \|\psi\|_{H^{1/2}(\Gamma)} \right)$$
$$\le C \left(\|g\|_{H^{3/2}(\partial\Omega)} + \|\varphi\|_{H^{3/2}(\Gamma)} + \|\psi\|_{H^{1/2}(\Gamma)} \right),$$

where C is a generic constant that depends on Ω, a_1, a_2. Let

$$\tilde{u}(x) = \begin{cases} \tilde{u}_1(x), & x \in \Omega_1, \\ \tilde{u}_2(x), & x \in \Omega_2. \end{cases}$$

Obviously, $\tilde{u}(x) \in X$. In addition, by [53,54] we know that the equation

$$-\nabla \cdot (a_i \nabla v) + b_i v = f_i + \nabla \cdot (a_i \nabla \tilde{u}_i) - b_i \tilde{u}_i, \quad \text{in } \Omega_i, \ i = 1,2,$$
$$[\![a\nabla v \cdot \mathbf{n}]\!] = 0, \quad \text{on } \Gamma,$$
$$[\![v]\!] = 0, \quad \text{on } \Gamma,$$
$$v = 0, \quad \text{on } \partial\Omega,$$

has a unique solution $v \in X$ and v satisfies the estimate

$$\|v\|_X \leq C \left(\|f_1 + \nabla \cdot (a_1 \nabla \tilde{u}_1) - b_1 \tilde{u}_1\|_{L^2(\Omega_1)} + \|f_2 + \nabla \cdot (a_2 \nabla \tilde{u}_2) - b_2 \tilde{u}_2\|_{L^2(\Omega_2)} \right)$$
$$\leq C \left(\|f_1\|_{L^2(\Omega_1)} + \|f_2\|_{L^2(\Omega_2)} + \|\tilde{u}_1\|_{H^2(\Omega_1)} + \|\tilde{u}_2\|_{H^2(\Omega_2)} \right).$$

Finally, we obtain that $u = v + \tilde{u}$ solves the problem (2.4) and that

$$\|u\|_X \leq \|v\|_X + \|\tilde{u}\|_X$$
$$\leq C \left(\|f_1\|_{L^2(\Omega_1)} + \|f_2\|_{L^2(\Omega_2)} + \|\tilde{u}\|_X \right)$$
$$\leq C \left(\|f_1\|_{L^2(\Omega_1)} + \|f_2\|_{L^2(\Omega_2)} + \|g\|_{H^{3/2}(\partial\Omega)} + \|\varphi\|_{H^{3/2}(\Gamma)} + \|\psi\|_{H^{1/2}(\Gamma)} \right)$$
$$\leq C \left(\|f_1\|_{L^2(\Omega_1)} + \|f_2\|_{L^2(\Omega_2)} + \|g\|_{H^2(\partial\Omega)} + \|\varphi\|_{H^2(\Gamma)} + \|\psi\|_{H^1(\Gamma)} \right),$$

where C is a generic constant. The proof is completed. $\square$

With these results, we are able to provide the proof of Theorem 4.1.

Proof of Theorem 4.1. Lemma 6.4 implies the existence and the uniqueness of solution u^*. Let

$$(u_{1,m}, u_{2,m}) \in (\mathcal{H}_{1,m}, \mathcal{H}_{2,m})$$

be a minimizer of the Lipschitz regularized loss $\text{Loss}_m(\cdot; \lambda, \lambda_m^R)$ (3.2). Again, by Lemma 6.4, we have that

$$\|u_{1,m} - u^*\|_{H^2(\Omega_1)} + \|u_{2,m} - u^*\|_{H^2(\Omega_2)} \leq C \Big(\|\mathcal{L}_1[u_{1,m}] - f_1\|_{L^2(\Omega_1)} + \|\mathcal{L}_2[u_{2,m}] - f_2\|_{L^2(\Omega_2)}$$
$$+ \|u_{2,m} - g\|_{H^2(\partial\Omega)} + \|u_{2,m} - u_{1,m} - \varphi\|_{H^2(\Gamma)}$$
$$+ \|a\nabla u_{2,m} \cdot \mathbf{n} - a\nabla u_{1,m} \cdot \mathbf{n} - \psi\|_{H^2(\Gamma)} \Big).$$

Finally, by the second term of Lemma 6.3, we conclude that with probability one over iid samples,

$$\lim_{m_{r_1} \to \infty} u_{1,m} = u^*, \quad \text{in } H^2(\Omega_1), \qquad \lim_{m_{r_1} \to \infty} u_{2,m} = u^*, \quad \text{in } H^2(\Omega_2),$$

which completes the proof. $\square$

7 Summary

The main contribution of this paper is to perform the convergence analysis of the neural network method for solving linear second-order elliptic interface problems. It is proved that the neural network sequence converges to the unique solution to the interface problem in H^2. Numerical results are presented to show agreement with the theoretical findings. This result advanced the mathematical foundations of the deep learning-based solver of PDEs.

To complete the proof, we first derive a Lipschitz regularized empirical loss from the probabilistic space filling arguments [43] to bound the expected PINN loss and then show that the expected PINN loss at the minimizers of the Lipschitz regularized empirical loss converges to zero. Finally, we demonstrate that the minimizers of the Lipschitz regularized empirical losses converge to the solution to the interface problem uniformly as the number of training samples grows in H^2 and conclude the main theorem.

There are several interesting further research directions. The landscape of non-convex objective functions and the stochastic gradient optimization process remain open. We would like to further investigate such problems and quantify the optimization error of solving elliptic interface problems using neural networks in the future. In addition, further error analysis to provide a more restrictive error bound is another interesting direction.

Acknowledgments

References

[1] A. Pinkus, Approximation theory of the MLP model in neural networks, Acta Numerica 8 (1999) 143–195.

[2] J. Darbon, T. Meng, On some neural network architectures that can represent viscosity solutions of certain high dimensional Hamilton–Jacobi partial differential equations, Journal of Computational Physics 425 (2021) 109907.

[3] A. Zhu, P. Jin, Y. Tang, Approximation capabilities of measure-preserving neural networks, Neural Networks 147 (2022) 72–80.

[4] E. Weinan, J. Han, A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, Communications in Mathematics and Statistics 5 (4) (2017) 349–380.

[5] M. Raissi, P. Perdikaris, G. 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 (2019) 686–707.

[6] H. Lee, I. S. Kang, Neural algorithm for solving differential equations, Journal of Computational Physics 91 (1) (1990) 110–131.

[7] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9 (5) (1998) 987–1000.

[8] K. Rudd, S. Ferrari, A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks, Neurocomputing 155 (2015) 277–285.

[9] G. Pang, L. Lu, G. E. Karniadakis, fPINNs: Fractional physics-informed neural networks, SIAM Journal on Scientific Computing 41 (4) (2019) A2603–A2626.

[10] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, Journal of Computational Physics 411 (2020) 109409.

[11] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deeponets, Science Advances 7 (40) (2021) eabi8605.

[12] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81.

[13] T. Y. Hou, Z. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele–Shaw flow, Journal of Computational Physics 134 (2) (1997) 236–252.

[14] B. Lu, Y. Zhou, M. Holst, J. McCammon, Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications, Commun. Comput. Phys. 3 (5) (2008) 973–1009.

[15] Y. Liu, M. Sussman, Y. Lian, M. Y. Hussaini, A moment-of-fluid method for diffusion equations on irregular domains in multi-material systems, Journal of Computational Physics 402 (2020) 109017.

[16] L. Wang, H. Zheng, X. Lu, L. Shi, A Petrov-Galerkin finite element interface method for interface problems with Bloch-periodic boundary conditions and its application in phononic crystals, Journal of Computational Physics 393 (2019) 117–138.

[17] Y. Chen, S. Hou, X. Zhang, A bilinear partially penalized immersed finite element method for elliptic interface problems with multi-domain and triple-junction points, Results in Applied Mathematics 8 (2020) 100100.

[18] N. Ji, T. Liu, J. Xu, L. Q. Shen, B. Lu, A finite element solution of lateral periodic Poisson–Boltzmann model for membrane channel proteins, International Journal of Molecular Sciences 19 (3) (2018) 695.

[19] R. Massjung, An unfitted discontinuous Galerkin method applied to elliptic interface problems, SIAM Journal on Numerical Analysis 50 (6) (2012) 3134–3162.

[20] R. I. Saye, Efficient multigrid solution of elliptic interface problems using viscosity-upwinded local discontinuous Galerkin methods, Communications in Applied Mathematics and Computational Science 14 (2) (2019) 247–283.

[21] R. J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM Journal on Numerical Analysis 31 (4) (1994) 1019–1044.

[22] S. Hou, X.-D. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, Journal of Computational Physics 202 (2) (2005) 411–445.

[23] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[24] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains–the Voronoi interface method, Journal of Computational Physics 298 (2015) 747–765.

[25] A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A

generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, Communications in Computational Physics 28 (5) (2020) 2002–2041.

[26] W. Li, X. Xiang, Y. Xu, Deep domain decomposition method: Elliptic problems, in: Mathematical and Scientific Machine Learning, PMLR, 2020, pp. 269–286.

[27] C. He, X. Hu, L. Mu, A mesh-free method using piecewise deep neural network for elliptic interface problems, arXiv preprint arXiv:2005.04847 (2020).

[28] S. Wu, B. Lu, INN: Interfaced neural networks as an accessible meshless approach for solving interface PDE problems, Journal of Computational Physics 470 (2022) 111588.

[29] Y. Shin, J. Darbon, G. Em Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, Communications in Computational Physics 28 (5) (2020) 2042–2074.

[30] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, IMA Journal of Numerical Analysis (2021).

[31] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, IMA Journal of Numerical Analysis (2022).

[32] Y. Shin, Z. Zhang, G. E. Karniadakis, Error estimates of residual minimization using neural networks for linear PDEs, arXiv preprint arXiv:2010.08019 (2020).

[33] Z. Hu, A. D. Jagtap, G. E. Karniadakis, K. Kawaguchi, When do extended physics-informed neural networks (XPINNs) improve generalization?, arXiv preprint arXiv:2109.09444 (2021).

[34] T. Luo, H. Yang, Two-layer neural networks for partial differential equations: Optimization and generalization theory, arXiv preprint arXiv:2006.15733 (2020).

[35] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Transactions on Information Theory 39 (3) (1993) 930–945.

[36] Y. Jiao, Y. Lai, D. Li, X. Lu, Y. Wang, J. Z. Yang, Convergence analysis for the PINNs, arXiv preprint arXiv:2109.01780 (2021).

[37] Y. Lu, J. Lu, M. Wang, A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations, in: Conference on Learning Theory, PMLR, 2021, pp. 3196–3241.

[38] J. Berner, P. Grohs, A. Jentzen, Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations, SIAM Journal on Mathematics of Data Science 2 (3) (2020) 631–657.

[39] P. L. Lagari, L. H. Tsoukalas, S. Safarkhani, I. E. Lagaris, Systematic construction of neural forms for solving partial differential equations inside rectangular domains, subject to initial, boundary and interface conditions, International Journal on Artificial Intelligence Tools 29 (05) (2020) 2050009.

[40] J. Yu, L. Lu, X. Meng, G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse pde problems, Computer Methods in Applied Mechanics and Engineering 393 (2022) 114823.

[41] J. L. Lions, E. Magenes, Non-Homogeneous Boundary Value Problems and Applications: Vol. 1, Vol. 181, Springer Science & Business Media, 2012.

[42] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey, Journal of Machine Learning Research 18 (153) (2018) 1–43.

[43] J. Calder, Consistency of lipschitz learning with infinite unlabeled data and finite labeled data, SIAM Journal on Mathematics of Data Science 1 (4) (2019) 780–812.

[44] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems 2 (4) (1989) 303–314.

[45] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks 3 (5) (1990) 551 – 560.

[46] I. Gühring, M. Raslan, Approximation rates for neural networks with encodable weights in smoothness spaces, Neural Networks 134 (2021) 107–130.

[47] T. De Ryck, S. Lanthaler, S. Mishra, On the approximation of functions by tanh neural networks, Neural Networks 143 (2021) 732–750.

[48] Z. Shen, H. Yang, S. Zhang, Neural network approximation: Three hidden layers are enough, Neural Networks 141 (2021) 160–173.

[49] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.

[50] Z. Wang, Z. Zhang, A mesh-free method for interface problems using the deep learning approach, Journal of Computational Physics 400 (2020) 108963.

[51] P. Grisvard, Elliptic problems in nonsmooth domains, SIAM, 2011.

[52] V. Girault, P.-A. Raviart, Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms, Vol. 5, Springer Science & Business Media, 2012.

[53] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (3) (1970) 207–213.

[54] R. Bruce Kellogg, On the Poisson equation with intersecting interfaces, Applicable Analysis 4 (2) (1974) 101–129.