

# SOLVING POISSON PROBLEMS IN POLYGONAL DOMAINS WITH SINGULARITY ENRICHED PHYSICS INFORMED NEURAL NETWORKS\*

TIANHAO HU<sup>†</sup>, BANGTI JIN<sup>†</sup>, AND ZHI ZHOU<sup>‡</sup>

**Abstract.** Physics-informed neural networks (PINNs) are a powerful class of numerical solvers for partial differential equations, employing deep neural networks with successful applications across a diverse set of problems. However, their effectiveness is somewhat diminished when addressing issues involving singularities, such as point sources or geometric irregularities, where the approximations they provide often suffer from reduced accuracy due to the limited regularity of the exact solution. In this work, we investigate PINNs for solving Poisson equations in polygonal domains with geometric singularities and mixed boundary conditions. We propose a novel singularity enriched PINN, by explicitly incorporating the singularity behavior of the analytic solution, e.g., corner singularity, mixed boundary condition, and edge singularities, into the ansatz space, and present a convergence analysis of the scheme. We present extensive numerical simulations in two and three dimensions to illustrate the efficiency of the method, and also a comparative study with several existing neural network based approaches.

**Key words.** Poisson equation, corner singularity, edge singularity, physics informed neural network, singularity enrichment

**MSC code.** 65N12

**DOI.** 10.1137/23M1601195



See reproducibility of computational results at end of the article.

**1. Introduction.** Partial differential equations (PDEs) represent a broad class of mathematical models that occupy a vital role in physics, science, and engineering. Many traditional PDE solvers have been developed, e.g., the finite difference method, the finite element method, and the finite volume method. They have been maturely developed over past decades, and efficient implementations and mathematical guarantees are also available. In the last few years, motivated by the great successes in diverse areas (computer vision, speech recognition and natural language processing, etc.), neural solvers for PDEs using deep neural networks (DNNs) have received much attention [44]. The list of neural solvers includes physics informed neural networks (PINNs) [56], the deep Ritz method (DRM) [65], the deep Galerkin method [60], the weak adversarial network [67], and the deep least-squares method [12], to name a few. Compared with traditional methods, neural PDE solvers have shown very promising results in several direct and inverse problems [38, 23, 17].

\*Submitted to the journal's Machine Learning Methods for Scientific Computing section September 13, 2023; accepted for publication (in revised form) April 11, 2024; published electronically July 30, 2024.

<https://doi.org/10.1137/23M1601195>

**Funding:** The work of the second author is supported by UK EPSRC grant EP/V026259/1, the Hong Kong RGC General Research Fund (project 14306423), and a start-up fund from the Chinese University of Hong Kong. The work of the third author is supported by the Hong Kong Research Grants Council (15303021) and an internal grant of Hong Kong Polytechnic University (project P0038888, work program 1-ZVX3).

<sup>†</sup>Department of Mathematics, Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, People's Republic of China (1155202953@link.cuhk.edu.hk, bangti.jin@gmail.com, b.jin@cuhk.edu.hk).

<sup>‡</sup>Department of Applied Mathematics, Hong Kong Polytechnic University, Kowloon, Hong Kong, People's Republic of China (zhizhou@polyu.edu.hk).

In these neural solvers, one employs DNNs as ansatz functions to approximate the solution to the PDE either in strong, weak, or Ritz forms. Existing approximation theory of DNNs [31] indicates that the accuracy of DNN approximations depends crucially on the Sobolev regularity of the solution (also suitable stability of the mathematical formulation). Thus, these methods might be ineffective or even fail completely when applied to problems with irregular solutions [63, 43], e.g., convection-dominated problems, transport problems, high-frequency wave propagation, problems with geometric singularities (cracks/corner singularity), and singular sources. All of these settings lead to either strong directional behavior, solution singularities, or highly oscillatory behavior, which are challenging for standard DNNs to approximate effectively.

Thus, there is an imperative need to develop neural solvers for PDEs with non-smooth solutions. Several recent efforts have been devoted to addressing the issue, including self-adaptive PINN (SAPINN) [33], failure-informed PINN (FIPINN) [27, 28], and singularity splitting DRM (SSDRM) [32]. SAPINN extends the PINN by splitting out the regions with singularities and then setting different weights to compensate the effect of singular regions. FIPINN [28] is inspired by the classical adaptive FEM, using the PDE residual as the indicator to aid judicious selection of sampling points for training. SSDRM [32] exploits analytic insights into the exact solution by approximating only the regular part using DNNs but extracting the singular part explicitly. These methods have shown remarkable performance for problems with significant singularities. One prime example is point sources, whose solutions involve localized singularities that can be extracted using fundamental solutions. See section 2.3 for further discussions about these methods.

In this work, we continue this line of research for Poisson problems on polygonal domains, which involve geometric singularities, including corners and mixed boundary conditions in the two-dimensional (2D) case and edges in the three-dimensional (3D) case. This represents an important setting in practical applications that has received enormous attention; see [30, 41, 42, 50] for the solution theory. We shall develop a class of effective neural solvers for Poisson problems with geometric singularities based on the idea of singularity enrichment, building on known analytic insights of the problems, and we call the proposed method singularity enriched PINN (SEPINN).

**1.1. Problem setting.** First, we state the mathematical formulation of the problem. Let  $\Omega \in \mathbb{R}^d$  ( $d = 2, 3$ ) be an open, bounded polygonal domain with a boundary  $\partial\Omega$ , and let  $\Gamma_D$  and  $\Gamma_N$  be a partition of  $\partial\Omega$  such that  $\Gamma_D \cup \Gamma_N = \partial\Omega$  and  $\Gamma_D \cap \Gamma_N = \emptyset$ , with a nonempty  $\Gamma_D$  (i.e., Lebesgue measure  $|\Gamma_D| \neq 0$ ). Let  $n$  denote the unit outward normal vector to  $\partial\Omega$ , and let  $\partial_n u$  denote taking the outward normal derivative. Given a source  $f \in L^2(\Omega)$ , consider the following Poisson problem:

$$(1.1) \quad \begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ \partial_n u = 0 & \text{on } \Gamma_N. \end{cases}$$

We focus on the zero boundary conditions, and nonzero ones can be transformed to (1.1) using the trace theorem. Due to the existence of corners, cracks, or edges in  $\Omega$ , the solution  $u$  of problem (1.1) typically exhibits singularities, even if  $f$  is smooth. The presence of singularities in the solution  $u$  severely deteriorates the accuracy of standard numerical methods for constructing approximations, including neural solvers, and more specialized techniques are needed in order to achieve high efficiency. Next we briefly review existing techniques for resolving the singularities.

In the 2D case, there are several classes of traditional numerical solvers based on FEM, including singularity representation based approaches [24, 61, 13], mesh grading [57, 3, 2], generalized FEM [26] and adaptive FEM [29], etc. These methods require different amounts of knowledge about the analytic solution. The methods in the first class exploit a singular representation of the solution  $u$  as a linear combination of singular and regular parts [30, 18] and can be further divided into four groups.

- (i) The singular function method augments singular functions to both trial and test spaces [24, 61]. However, the convergence of the coefficients (a.k.a. stress intensity factors) sometimes is poor [22], which may lead to low accuracy.
- (ii) The dual singular function method [10] employs the dual singular function to extract the coefficients as a postprocessing strategy of FEM, which can also achieve the theoretical rate in practical computation.
- (iii) The singularity splitting method [13] splits the singular part from the solution  $u$  and approximates the smooth part with the Galerkin FEM, and enjoys  $H^1(\Omega)$  and  $L^2(\Omega)$  error estimates. It can improve the accuracy of the approximation, and the stress intensity factor can also be obtained from the extraction formula; cf. (3.6).
- (iv) The singular complement method [4] is based on an orthogonal decomposition of the solution  $u$  into a singular part and a regular part, by augmenting the FEM trial space with specially designed singular functions.

For 3D problems with edges, the singular functions belong to an infinite dimensional space and their coefficients are functions defined along edges [30]. Thus, their computation involves approximating functions defined along edges, and there are relatively few numerical methods, and numerical investigations are strikingly lacking. The methods developed for the 2D case do not extend directly to the 3D case. In fact, in several existing studies, numerical algorithms and error analysis have been provided, but the methods are nontrivial to implement [55, 61]. For example, the approach in [55] requires evaluating a few dozens of highly singular integrals at each step, which may lead to serious numerical issues.

**1.2. Our contributions.** In this work, by building analytic knowledge of the problem into numerical schemes, we construct a novel numerical method using PINN to solve Poisson problems with geometric singularities. The key analytic insight is that the solution  $u$  has a singular function representation as a linear combination of a singular function  $S$  and a regular part  $w$  [18, 19, 30, 40]:  $u = S + w$  with  $w \in H^2(\Omega)$ . The singular function  $S$  is determined by the domain  $\Omega$ , truncation functions, and their coefficients. Using this fact, we develop, analyze, and test SEPINN and make the following contributions:

- (i) develop a novel class of SEPINNs for corner singularities, mixed boundary conditions, and edge singularity for the Poisson problem;
- (ii) provide error bounds for the SEPINN approximation;
- (iii) present numerical experiments for multiple scenarios, including 3D problems with edge singularities and the eigenvalue problem on an L-shaped domain, to illustrate the flexibility and accuracy of SEPINN. We also include a comparative study with existing approaches.

To the best of our knowledge, this is the first work systematically exploring the use of singularity enrichment in a neural PDE solver.

The rest of the paper is organized as follows. In section 2 we recall preliminaries on DNNs and its use in PINNs. Then we develop the SEPINN in section 3 for the 2D case (corner singularity and mixed boundary conditions) and the 3D case (edge singularity)

separately. In section 4, we discuss the convergence analysis of SEPINN. In section 5, we present numerical experiments to illustrate the performance of SEPINN, including a comparative study with PINN and its variants, and give further discussions in section 6.

## 2. Preliminaries.

**2.1. Deep neural networks.** We employ standard fully connected feedforward DNNs, i.e., functions  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ , with the DNN parameters  $\theta \in \mathbb{R}^{N_\theta}$  ( $N_\theta$  is the dimensionality of the DNN parameters). Given a sequence of integers  $\{n_\ell\}_{\ell=0}^L$ , with  $n_0 = d$  and  $n_L = 1$ ,  $f_\theta(x)$  is defined recursively by  $f^{(0)} = x$ ,  $f^{(\ell)} = \varrho(A_\ell f^{(\ell-1)} + b_\ell)$ ,  $\ell = 1, 2, \dots, L-1$ ,  $f_\theta(x) := f^{(L)}(x) = A_L f^{(L-1)} + b_L$ , where  $A_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$  and  $b_\ell \in \mathbb{R}^{n_\ell}$ ,  $\ell = 1, 2, \dots, L$ , are the weight matrix and bias vector at the  $\ell$ th layer. The nonlinear activation function  $\varrho : \mathbb{R} \rightarrow \mathbb{R}$  is applied componentwise to a vector. The integer  $L$  is called the depth and  $W := \max\{n_\ell, \ell = 0, 1, \dots, L\}$  the width of the DNN. The set of parameters  $\{A_\ell, b_\ell\}_{\ell=1}^L$  of the DNN is trainable and stacked into a big vector  $\theta$ .  $f^{(0)}$  is called the input layer;  $f^{(\ell)}$ ,  $\ell = 1, 2, \dots, L-1$ , are called the hidden layer; and  $f_\theta(x)$  is the output layer.

There are many possible choices of  $\varrho$ . The most frequently used one in computer vision is rectified linear unit (ReLU),  $\varrho(x) = \max(x, 0)$ . However, it is not smooth enough for PINN, since PINN requires thrice differentiability of  $\varrho$ : two spatial derivatives in the loss, and another derivative in the DNN parameters  $\theta$  (for the optimizer). For neural PDE solvers, the hyperbolic tangent  $\varrho(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  and logistic/ sigmoid  $\varrho(x) = \frac{1}{1+e^{-x}}$  are often used [56, 17]. We employ the hyperbolic tangent. We denote the collection of DNN functions of depth  $L$ , with  $N_\theta$  nonzero parameters, and each of the parameters bounded by  $B_\theta$ , with the activation function  $\varrho$  by  $\mathcal{N}_\varrho(L, N_\theta, B_\theta)$ , i.e.,  $\mathcal{N}_\varrho(L, N_\theta, B_\theta) = \{w_\theta : w_\theta \text{ has a depth } L, |\theta|_0 \leq N_\theta, |\theta|_{\ell^\infty} \leq B_\theta\}$ , where  $|\cdot|_{\ell^0}$  and  $|\cdot|_{\ell^\infty}$  denote the number of nonzero entries in and the maximum norm of a vector, respectively. We also use the notation  $\mathcal{A}$  to denote this collection of functions.

**2.2. Physics informed neural networks.** PINNs [56] represent one popular neural solver based on the principle of PDE residual minimization. For problem (1.1), the continuous loss  $\mathcal{L}(u)$  is given by

$$(2.1) \quad \mathcal{L}_\sigma(u) = \|\Delta u + f\|_{L^2(\Omega)}^2 + \sigma_d \|u\|_{L^2(\Gamma_D)}^2 + \sigma_n \|\partial_n u\|_{L^2(\Gamma_N)}^2,$$

where the tunable penalty weights  $\sigma_d, \sigma_n > 0$  are to approximately enforce the boundary conditions, and  $\sigma = (\sigma_d, \sigma_n)$ . We approximate the solution  $u$  by an element  $u_\theta \in \mathcal{A}$ , and then discretize relevant integrals using quadrature, e.g., the Monte Carlo method. Let  $U(D)$  be the uniform distribution over a set  $D$ , and let  $|D|$  denote its Lebesgue measure. Then we can rewrite the loss  $\mathcal{L}(u_\theta)$  as

$$\begin{aligned} \mathcal{L}_\sigma(u_\theta) = & |\Omega| \mathbb{E}_{U(\Omega)}[(\Delta u_\theta(X) + f(X))^2] + \sigma_d |\Gamma_D| \mathbb{E}_{U(\Gamma_D)}[(u_\theta(Y))^2] \\ & + \sigma_n |\Gamma_N| \mathbb{E}_{U(\Gamma_N)}[(\partial_n u_\theta(Z))^2], \end{aligned}$$

where  $\mathbb{E}_\nu$  takes expectation with respect to a distribution  $\nu$ . Let the sampling points  $\{X_i\}_{i=1}^{N_r}$ ,  $\{Y_j\}_{j=1}^{N_d}$ , and  $\{Z_k\}_{k=1}^{N_n}$  be identically and independently distributed (i.i.d.), uniformly on  $\Omega$ ,  $\Gamma_D$ , and  $\Gamma_N$ , respectively, i.e.,  $\{X_i\}_{i=1}^{N_r} \sim U(\Omega)$ ,  $\{Y_j\}_{j=1}^{N_d} \sim U(\Gamma_D)$ , and  $\{Z_k\}_{k=1}^{N_n} \sim U(\Gamma_N)$ . Then the empirical loss  $\hat{\mathcal{L}}_\sigma(u_\theta)$  is given by

$$\hat{\mathcal{L}}_\sigma(u_\theta) = \frac{|\Omega|}{N_r} \sum_{i=1}^{N_r} (\Delta u_\theta(X_i) + f(X_i))^2 + \frac{\sigma_d |\Gamma_D|}{N_d} \sum_{j=1}^{N_d} (u_\theta(Y_j))^2 + \frac{\sigma_n |\Gamma_N|}{N_n} \sum_{k=1}^{N_n} (\partial_n u_\theta(Z_k))^2.$$

Note that the resulting optimization problem  $\widehat{\mathcal{L}}_{\sigma}(u_{\theta})$  over  $\mathcal{A}$  is well posed due to the box constraint on the DNN parameters  $\theta$ , i.e.,  $|\theta|_{\ell^{\infty}} \leq R$  for suitable  $R$ , which induces a compact set in  $\mathbb{R}^{N_{\theta}}$ . Meanwhile, the empirical loss  $\widehat{\mathcal{L}}_{\sigma}(u_{\theta})$  is continuous in  $\theta$ , when  $\varrho$  is smooth. In the absence of the box constraint, the optimization problem might not have a finite minimizer.

The loss  $\widehat{\mathcal{L}}_{\sigma}(u_{\theta})$  is minimized with respect to the DNN parameters  $\theta$ . This is often achieved by gradient type algorithms, e.g., Adam [39] or limited memory BFGS [11], which return an approximate minimizer  $\theta^*$ . The DNN approximation to the PDE solution  $u$  is given by  $u_{\theta^*}$ . Note that the major computational effort, e.g., gradient of  $\widehat{\mathcal{L}}_{\sigma}(u_{\theta})$  with respect to the DNN parameters  $\theta$  and the DNN  $u_{\theta}(x)$  with respect to the input  $x$ , can both be computed efficiently via automatic differentiation [7], which is available in many software platforms, e.g., PyTorch or Tensorflow. Thus, the method is very flexible and easy to implement and is applicable to a wide range of direct and inverse problems for PDEs [38].

The population loss  $\mathcal{L}_{\sigma}(u_{\theta})$  and empirical loss  $\widehat{\mathcal{L}}_{\sigma}(u_{\theta})$  have different minimizers, due to the presence of quadrature errors. The analysis of these errors is known as generalization error analysis in statistical learning theory [1]. The theoretical analysis of PINNs has been investigated in several works under different settings [58, 34, 51, 48, 21, 20]. Important issues addressed in these works include the existence of a neural network such that the population/empirical loss is small, connection between the error of the approximate solution with the population loss (via suitable stability of the formulation), and the gap between the population loss and training loss. These roughly correspond to approximation theory of DNNs in Sobolev spaces, coercivity of the formulation, and (Monte Carlo) quadrature error. These mathematical theories require that the solutions to the problems be smooth, e.g.,  $C^2(\bar{\Omega})$ , in order to achieve consistency [58] and even stronger regularity for convergence rates [34]. Such conditions unfortunately cannot be met for problem (1.1), due to the inherently limited solution regularity. Thus, it is not a priori clear that one can successfully apply PINNs to problem (1.1). This is also confirmed by the numerical experiments in section 5. In practice, despite the ease of implementation, the standard PINN may fail to converge [43, 63]. Indeed, there are several failure modes, including the curse of dimensionality, low regularity, discontinuities, etc. The concerned class of problems involves solution singularity, which represents one typical failure mode of PINN. In this work, we shall develop an effective strategy to overcome the challenge.

**2.3. Two existing methods.** Now we describe two PINN approaches to address the singularity, i.e., SAPINN and FIPINN, for the Dirichlet problem of the Laplace equation with a singular source  $f$  with singularities located within a subdomain  $\Omega_s \subset \Omega$ .

In SAPINN [33], one treats the PDE residuals in the regions  $\Omega_s$  and  $\Omega \setminus \Omega_s$  separately and further splits the loss into

$$L(u_{\theta}) = \lambda_s L_s(u_{\theta}) + \lambda_r L_r(u_{\theta}) + \lambda_b L_b(u_{\theta})$$

with  $L_s(u_{\theta}) = \|\Delta u_{\theta} + f\|_{L^2(\Omega_s)}^2$ ,  $L_r(u_{\theta}) = \|\Delta u_{\theta} + f\|_{L^2(\Omega \setminus \Omega_s)}^2$  and  $L_b(u_{\theta}) = \|u_{\theta}\|_{L^2(\partial\Omega)}^2$ , and  $\lambda_i$ ,  $i \in \{r, s, b\}$ , being positive weights. The partition of  $\Omega$  into  $\Omega_s$  and  $\Omega \setminus \Omega_s$  and using different penalty parameters allow better compensation for the singular behavior of  $u$  in the subdomain  $\Omega_s$ , in a manner similar to weighted Sobolev spaces. The experimental results in [33] indicate that the method can improve the accuracy of the approximation, but the increase in the number of hyperparameters (i.e.,  $\lambda_i$ s) also complicates the optimization process, due to the need for self-adaptation during training, which in turn makes the overall training lengthier.

FIPINN [28] employs adaptive sampling. Let  $\mathcal{Q}(x) : \Omega \rightarrow [0, \infty)$  be a performance function, e.g., the PDE residual  $r(x; \theta) = (\Delta u_\theta(x) + f(x))^2$  and its spatial gradient  $\nabla_x r(x; \theta)$ , and let  $\Omega_{\mathcal{F}} := \{x : \mathcal{Q}(x) > \epsilon_r\}$  be the failure region (with tolerance  $\epsilon_r$ ). Then over  $\Omega_{\mathcal{F}}$ , one can define the failure probability  $P_{\mathcal{F}} = \int_{\Omega_{\mathcal{F}}} \omega(x) dx$ , where  $\omega(x)$  is the prior distribution of the sampling point  $x$ . The failure probability  $P_{\mathcal{F}}$  may serve as an error indicator for training set refinement, similar to the classical adaptive FEM. Using  $P_{\mathcal{F}}$ , new candidate collocation points can be generated to form the loss in the hope of enhancing PINN's performance after retraining (simultaneously for estimating  $P_{\mathcal{F}}$ ). In [28], a truncated Gaussian is taken for importance sampling in order to approximatively determine  $P_{\mathcal{F}}$  and to produce new training points. These steps are looped until a suitable stopping criterion is met.

Note that to compensate the impact of solution singularity, SAPINN and FIPINN follow different strategies: SAPINN adaptively updates the weights assigned to the losses in the subdomains, whereas FIPINN employs the residual as an a posteriori error estimator to adaptively generate sampling points in the empirical distribution.

**3. Singularity enriched PINN.** Now we develop a class of SEPINN for solving Poisson problems with geometric singularities, including 2D problems with mixed boundary conditions or on polygonal domains, and 3D problems with edge singularity. The key is the singular function representation. We discuss the 2D case in section 3.1 and the more involved 3D case in section 3.2. The approach applies also to the modified Helmholtz equation, which we describe in the supplementary material (M160119\_SuppMat.pdf [local/web 399KB]).

### 3.1. Two-dimensional problem.

**3.1.1. Singular function representation.** First we develop a singular function representation in the 2D case. We determine the analytic structure of the solution  $u$  of problem (1.1) using the Fourier method. Consider a vertex  $\mathbf{v}_j$  of the polygonal domain  $\Omega$  with an interior angle  $\omega_j$ . We denote by  $(r_j, \theta_j)$  the local polar coordinate of the vertex  $\mathbf{v}_j$  so that the interior angle  $\omega_j$  is spanned counterclockwise by two rays  $\theta_j = 0$  and  $\theta_j = \omega_j$ . Then consider the local behavior of  $u$  near the vertex  $\mathbf{v}_j$ , or in the sector  $G_j = \{(r_j, \theta_j) : 0 < r_j < R_j, 0 < \theta_j < \omega_j\}$ . Let the edge overlapping with  $\theta_j = 0$  be  $\Gamma_{j1}$  and that with  $\theta_j = \omega_j$  be  $\Gamma_{j2}$ ; cf. Figure 1(a) for a sketch. We employ the system of orthogonal and complete set of basis functions  $\{\phi_{j,k}\}_{k=1}^\infty$  in  $L^2(0, \omega_j)$  in

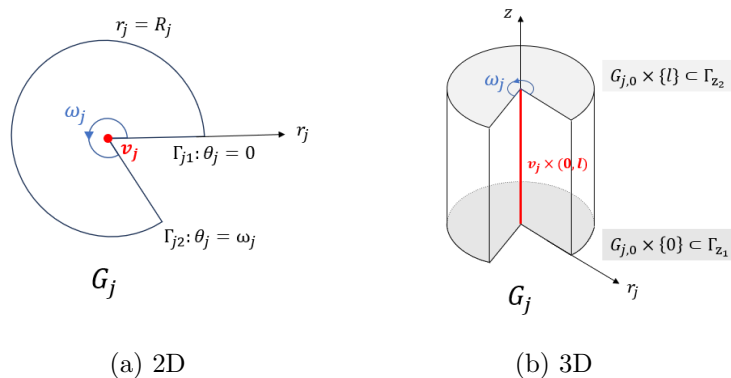


FIG. 1. A sketch of the local domain  $G_j$  in two and three dimensions. In the 3D case,  $G_{j,0}$  is defined by  $G_{j,0} = \{(x_j, y_j) \in \Omega_0 : 0 < r_j < R_j, 0 < \theta_j < \omega_j\}$ .

TABLE 1

Orthogonal basis functions  $\phi_{j,k}$  (dropping subscript  $j$ ),  $k \in \mathbb{N}$ , associated with the vertex  $\mathbf{v}_j$  in local polar coordinates, depending on the boundary conditions on  $\Gamma_{j_1}$  and  $\Gamma_{j_2}$ .

$\Gamma_{j_2} \setminus \Gamma_{j_1}$	Dirichlet	Neumann
Dirichlet	$\phi_k(\theta) = \sin \lambda_k \theta$ , $\lambda_k = \frac{k\pi}{\omega}$	$\phi_k(\theta) = \sin \lambda_k \theta$ , $\lambda_k = (k - \frac{1}{2}) \frac{\pi}{\omega}$
Neumann	$\phi_k(\theta) = \cos \lambda_k \theta$ , $\lambda_k = (k - \frac{1}{2}) \frac{\pi}{\omega}$	$\phi_k(\theta) = \cos \lambda_k \theta$ , $\lambda_k = (k - 1) \frac{\pi}{\omega}$

Table 1, where  $\lambda_{j,k}$  are the eigenvalues. More precisely, the pairs  $(\phi_{j,k}, \lambda_{j,k})$   $k = 1, \dots, \infty$ , are associated with the following Sturm–Liouville problem on the interval  $(0, \omega_j)$ :

$$-\phi''_{j,k} = \lambda_{j,k} \phi_{j,k} \quad \text{in } (0, \omega_j),$$

with suitable zero Dirichlet/Neumann boundary conditions, according to the boundary conditions (b.c.) on the edges  $\Gamma_{j_1}$  and  $\Gamma_{j_2}$  [54]. In the table, we have dropped the subscript  $j$  for notational simplicity. We denote the representation of  $u$  in the local polar coordinate by  $\tilde{u}(r_j, \theta_j)$ , i.e.,  $\tilde{u}(r_j, \theta_j) = u(r_j \cos \theta_j, r_j \sin \theta_j)$ .

To study the behavior of the solution  $u$  in the sector  $G_j$ , we assume  $\tilde{u}(R_j, \theta_j) = 0$  and  $|\tilde{u}(0, \theta_j)| < \infty$ . Since  $u$  and  $f$  belong to  $L^2(G_j)$ , they can be represented in Fourier series with respect to  $\{\phi_{j,k}\}_{k=1}^{\infty}$ :

$$(3.1) \quad u(x, y) = \tilde{u}(r_j, \theta_j) = \sum_{k=1}^{\infty} u_k(r_j) \phi_{j,k}(\theta_j) \quad \text{in } G_j,$$

$$(3.2) \quad f(x, y) = \tilde{f}(r_j, \theta_j) = \sum_{k=1}^{\infty} f_k(r_j) \phi_{j,k}(\theta_j) \quad \text{in } G_j$$

with  $u_k(r_j)$  and  $f_k(r_j)$  given respectively by  $u_k(r_j) = \frac{2}{\omega_j} \int_0^{\omega_j} \tilde{u}(r_j, \theta_j) \phi_{j,k}(\theta_j) d\theta_j$  and  $f_k(r_j) = \frac{2}{\omega_j} \int_0^{\omega_j} \tilde{f}(r_j, \theta_j) \phi_{j,k}(\theta_j) d\theta_j$ . Substituting (3.1) and (3.2) into (1.1) gives the following two-point boundary value problem for each vertex  $\mathbf{v}_j$ :

$$\begin{cases} -u''_{j,k} - r^{-1} u'_{j,k} + \lambda_{j,k}^2 r^{-2} u_{j,k} = f_k, & 0 < r < R_j, \\ |u_{j,k}(r)|_{r=0} < \infty, & u_{j,k}(r)|_{r=R_j} = 0. \end{cases}$$

Solving the ODE yields directly the closed-form expression

$$u_k(r_j) = c(r_j) r_j^{\lambda_{j,k}} + \frac{r_j^{-\lambda_{j,k}}}{2\lambda_{j,k}} \int_0^{r_j} f_k(\tau) \tau^{1+\lambda_{j,k}} d\tau$$

with the factor  $c(r_j)$  given by [54, (2.10)]

$$c(r_j) = \frac{1}{2\lambda_{j,k}} \int_{r_j}^{R_j} f_k(\tau) \tau^{1-\lambda_{j,k}} d\tau - \frac{1}{2\lambda_{j,k} R_j^{2\lambda_{j,k}}} \int_0^{R_j} f_k(\tau) \tau^{1+\lambda_{j,k}} d\tau.$$

Since the factor  $c(r_j)$  in front of  $r_j^{\lambda_{j,k}}$  is generally nonzero, and  $r_j^{\lambda_{j,k}} \notin H^2(G_j)$  for  $\lambda_{j,k} < 1$ , there exist singular terms in the representation (3.1). This shows the limited regularity of the solution  $u$ , which is the culprit of low efficiency of standard numerical schemes.

Next we list all singularity points and corresponding singular functions. Let  $\mathbf{v}_j$ ,  $j = 1, 2, \dots, M$ , be the vertices of  $\Omega$  whose interior angles  $\omega_j$ ,  $j = 1, 2, \dots, M$ , satisfy

TABLE 2

Singularity functions  $s_{j,i}$  (suppressing the subscript  $j$ ), depending on the boundary condition. The tuple  $(r_j, \theta_j)$  refers to the local polar coordinate of the vertex  $\mathbf{v}_j$ , and  $\mathbb{I}_j$  is an index set for leading singularities associated with  $\mathbf{v}_j$ .

$\Gamma_{j2} \setminus \Gamma_{j1}$	Dirichlet	Neumann
Dirichlet	$s_1(r, \theta_j) = r \frac{\pi}{\omega} \sin \frac{\pi\theta}{\omega},$ $\mathbb{I} = \{1\}$	$s_{\frac{1}{2}}(r, \theta) = r \frac{\pi}{2\omega} \cos \frac{\pi\theta}{2\omega}, \mathbb{I} = \{\frac{1}{2}\},$ if $\frac{\pi}{2} < \omega_j \leq \frac{3\pi}{2}$ <hr/> $s_{\frac{1}{2}}(r, \theta) = r \frac{\pi}{2\omega} \cos \frac{\pi\theta}{2\omega}$ and $s_{\frac{3}{2}}(r, \theta) = r \frac{3\pi}{2\omega} \cos \frac{3\pi\theta}{2\omega} \mathbb{I} = \{\frac{1}{2}, \frac{3}{2}\}$ if $\frac{3\pi}{2} < \omega \leq 2\pi$
Neumann	$s_{\frac{1}{2}}(r, \theta) = r \frac{\pi}{2\omega} \sin \frac{\pi\theta}{2\omega}, \mathbb{I} = \{\frac{1}{2}\},$ if $\frac{\pi}{2} < \omega \leq \frac{3\pi}{2}$ <hr/> $s_{\frac{1}{2}}(r, \theta) = r \frac{\pi}{2\omega} \sin \frac{\pi\theta}{2\omega}$ and $s_{\frac{3}{2}}(r, \theta) = r \frac{3\pi}{2\omega} \sin \frac{3\pi\theta}{2\omega} \mathbb{I} = \{\frac{1}{2}, \frac{3}{2}\},$ if $\frac{3\pi}{2} < \omega \leq 2\pi$	$s_1(r, \theta_j) = r \frac{\pi}{\omega} \cos \frac{\pi\theta}{\omega},$ $\mathbb{I} = \{1\}$

$$(3.3) \quad \begin{cases} \pi < \omega_j < 2\pi, & \text{b.c. doesn't change its type,} \\ \pi/2 < \omega_j < 2\pi, & \text{b.c. changes its type.} \end{cases}$$

Table 2 gives the index set  $\mathbb{I}_j$  and the associated singularity functions [14, p. 2637]. Note that  $s_{j,1} \in H^{1+\frac{\pi}{\omega}-\epsilon}(\Omega)$ ,  $s_{j,\frac{1}{2}} \in H^{1+\frac{\pi}{2\omega}-\epsilon}(\Omega)$ , and  $s_{j,\frac{3}{2}} \in H^{1+\frac{3\pi}{2\omega}-\epsilon}(\Omega)$  for small  $\epsilon > 0$ . Upon letting

$$\omega^* = \max_{1 \leq j \leq M} \hat{\omega}_j \quad \text{with } \hat{\omega}_j = \begin{cases} \omega_j, & \text{b.c. doesn't change its type at } \mathbf{v}_j, \\ 2\omega_j, & \text{b.c. changes its type at } \mathbf{v}_j, \end{cases}$$

the solution  $u$  belongs to  $H^{1+\frac{\pi}{\omega^*}-\epsilon}(\Omega)$ , just falling short of  $H^{1+\frac{\pi}{\omega^*}}(\Omega)$ . If  $\omega^* > \pi$ ,  $u$  fails to belong to  $H^2(\Omega)$ . Hence, it is imperative to develop techniques to resolve such singularities.

We employ a smooth cutoff function  $\eta_\rho$ , defined for  $\rho \in (0, 2]$  by [13, 15]

$$(3.4) \quad \eta_\rho(r_j) = \begin{cases} 1, & 0 < r_j < \frac{\rho R}{2}, \\ \frac{15}{16} \left( \frac{8}{15} - \left( \frac{4r_j}{\rho R} - 3 \right) + \frac{2}{3} \left( \frac{4r_j}{\rho R} - 3 \right)^3 - \frac{1}{5} \left( \frac{4r_j}{\rho R} - 3 \right)^5 \right), & \frac{\rho R}{2} \leq r_j < \rho R, \\ 0, & r_j \geq \rho R, \end{cases}$$

where  $R \in \mathbb{R}_+$  is a fixed number so that  $\eta_\rho$  vanishes identically on  $\partial\Omega$ . In practice, we take  $R$  to be small enough so that when  $i \neq j$ , the support of  $\eta_\rho(r_i)$  does not intersect with that of  $\eta_\rho(r_j)$ . By construction, we have  $\eta_\rho \in C^2([0, \infty))$ . This choice of  $\eta_\rho(r_j)$  preserves the zero boundary condition and also enjoys good regularity (and thus does not introduce extra singularities). In practice, any choice satisfying these desirable properties is valid. Then the solution  $u$  of problem (1.1) has the singular function representation [5, 18, 45]

$$(3.5) \quad u = w + \sum_{j=1}^M \sum_{i \in \mathbb{I}_j} \gamma_{j,i} \eta_{\rho_j}(r_j) s_{j,i}(r_j, \theta_j) \quad \text{with } w \in H^2(\Omega) \cap H_0^1(\Omega),$$



where the scalars  $\gamma_{j,i} \in \mathbb{R}$  are known as stress intensity factors and given by the following extraction formulas [30, Lemma 8.4.3.1]:

$$(3.6) \quad \gamma_{j,i} = \frac{1}{i\pi} \left( \int_{\Omega} f \eta_{\rho_j} s_{j,-i} dx + \int_{\Omega} u \Delta(\eta_{\rho_j} s_{j,-i}) dx \right),$$

where  $s_{j,-i}$  denotes dual singular functions of  $s_{j,i}$ . Specifically, if  $s_{j,i}(r_j, \theta_j) = r_j^{\frac{i\pi}{\omega_j}} \sin \frac{i\pi\theta_j}{\omega_j}$ , the dual function  $s_{j,-i}$  is given by  $s_{j,-i}(r_j, \theta_j) = r_j^{-\frac{i\pi}{\omega_j}} \sin \frac{i\pi\theta_j}{\omega_j}$  [14, p. 2639]. Moreover, the following regularity estimate on the regular part  $w$  holds [14, 30]:

$$(3.7) \quad \|w\|_{H^2(\Omega)} + \sum_{j=1}^M \sum_{i \in \mathbb{I}_j} |\gamma_{j,i}| \leq c \|f\|_{L^2(\Omega)}.$$

**3.1.2. Singularity enriched physics-informed neural network.** Now we propose SEPINN for problem (1.1), inspired by the representation (3.5). The idea of singularity enrichment/splitting has been widely used in the context of the FEM (see the introduction for details), and also in SSDRM [32], which splits out explicitly the singular part due to point/line sources. We discuss only the case with one singular function in  $\Omega$  satisfying the condition (3.3). The case of multiple singularities can be handled similarly. With  $S = \gamma \eta_{\rho} s$ , the regular part  $w$  satisfies

$$(3.8) \quad \begin{cases} -\Delta w = f + \gamma \Delta(\eta_{\rho} s) & \text{in } \Omega, \\ w = 0 & \text{on } \Gamma_D, \\ \partial_n w = 0 & \text{on } \Gamma_N, \end{cases}$$

where  $\gamma \in \mathbb{R}$  is unknown. Since  $w \in H^2(\Omega)$ , it can be well approximated using PINN. The parameter  $\gamma$  can be either learned together with  $w$  or extracted from  $w$  via (3.6).

Based on the principle of PDE residual minimization, the solution  $w^*$  of (3.8) and the exact parameter  $\gamma^*$  in (3.5) is a global minimizer of the following loss:

$$(3.9) \quad \mathcal{L}_{\sigma}(w, \gamma) = \|\Delta w + f + \gamma \Delta(\eta_{\rho} s)\|_{L^2(\Omega)}^2 + \sigma_d \|w\|_{L^2(\Gamma_D)}^2 + \sigma_n \|\partial_n w\|_{L^2(\Gamma_N)}^2,$$

where the penalty weights  $\sigma = (\sigma_d, \sigma_n) \in \mathbb{R}_+^2$  are tunable. Following section 2.2, we employ a DNN  $w_{\theta} \in \mathcal{A}$  to approximate  $w^* \in H^2(\Omega)$ , and we treat the parameter  $\gamma$  as a trainable parameter and learn it along with the DNN parameters  $\theta$ . This leads to an empirical loss,

$$(3.10) \quad \begin{aligned} \widehat{\mathcal{L}}_{\sigma}(w_{\theta}, \gamma) &= \frac{|\Omega|}{N_r} \sum_{i=1}^{N_r} (\Delta w_{\theta}(X_i) + f(X_i) + \gamma \Delta(\eta_{\rho} s)(X_i))^2 \\ &+ \sigma_d \frac{|\Gamma_D|}{N_d} \sum_{j=1}^{N_d} w_{\theta}^2(Y_j) + \sigma_n \frac{|\Gamma_N|}{N_n} \sum_{k=1}^{N_n} (\partial_n w_{\theta}(Z_k))^2, \end{aligned}$$

with i.i.d. sampling points  $\{X_i\}_{i=1}^{N_r} \sim U(\Omega)$ ,  $\{Y_j\}_{j=1}^{N_d} \sim U(\Gamma_D)$ , and  $\{Z_k\}_{k=1}^{N_n} \sim U(\Gamma_N)$ . Let  $(\widehat{\theta}^*, \widehat{\gamma}^*)$  be a minimizer of the empirical loss  $\widehat{\mathcal{L}}(w_{\theta}, \gamma)$ . Then  $w_{\widehat{\theta}^*} \in \mathcal{A}$  is the DNN approximation of the regular part  $w^*$ , and the approximation  $\widehat{u}$  to  $u$  is given by  $\widehat{u} = w_{\widehat{\theta}^*} + \widehat{\gamma}^* \eta_{\rho} s$ .

Now we discuss the training of the loss  $\widehat{\mathcal{L}}_{\sigma}(w_{\theta}, \gamma)$ . One can minimize  $\widehat{\mathcal{L}}_{\sigma}(w_{\theta}, \gamma)$  directly with respect to  $\theta$  and  $\gamma$ , which works reasonably. However, the DNN approximation  $\widehat{u}$  tends to have larger errors on the boundary  $\partial\Omega$  than in the domain  $\Omega$ , but the estimated  $\widehat{\gamma}^*$  is often accurate. Thus we adopt a two-stage training procedure.

---

**Algorithm 3.1.** SEPINN for 2D problems.

---

- 1: Set  $\sigma^{(1)}$ , and obtain the minimizer  $(\hat{\theta}^*, \hat{\gamma}^*)$  of the loss  $\hat{\mathcal{L}}_{\sigma^{(1)}}(w_\theta, \gamma)$ .
  - 2: Set  $k = 1$ ,  $\hat{\theta}_0^* = \hat{\theta}^*$ , and increasing factor  $q > 1$ .
  - 3: **while** Stopping condition not met **do**
  - 4:     Find a minimizer  $\hat{\theta}_k^*$  of the loss  $\hat{\mathcal{L}}_{\sigma^{(k)}}(w_\theta, \hat{\gamma}^*)$  (initialized to  $\hat{\theta}_{k-1}^*$ ).
  - 5:     Update  $\sigma$  by  $\sigma^{(k+1)} = q\sigma^{(k)}$ , and  $k \leftarrow k + 1$ .
  - 6: Output the SEPINN approximation  $\hat{u} = w_{\hat{\theta}_{k-1}^*} + \hat{\gamma}^* \eta s$ .
- 

- (i) At Stage 1, minimize the loss  $\hat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  for a fixed  $\sigma$ , and obtain the minimizer  $(\hat{\theta}^*, \hat{\gamma}^*)$ .
- (ii) At Stage 2, fix  $\gamma$  in  $\hat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  at  $\hat{\gamma}^*$ , and learn  $\theta$  via a path-following (PF) strategy [47, 32].

The estimate  $\hat{\gamma}^*$  depends on the choice of  $\sigma$ , but numerically it does not vary much with the choice. Moreover, the optimal  $\hat{\gamma}^*$  may vanish, i.e., the solution  $u$  is smooth.

Now we describe a PF strategy to update  $\sigma$ . We start with small values  $\sigma^{(1)} = \sigma$ . After each loop (i.e., finding one minimizer  $\hat{\theta}_k^*$ ), we update  $\sigma$  geometrically:  $\sigma^{(k+1)} = q\sigma^{(k)}$  with  $q > 1$ . By updating  $\sigma^{(k)}$ , the minimizer  $\hat{\theta}_k^*$  of the loss  $\hat{\mathcal{L}}_{\sigma^{(k)}}(w_\theta, \hat{\gamma}^*)$  also approaches that of problem (3.9), and the PF strategy enforces the boundary conditions progressively, which is beneficial to obtain good approximations, since when  $\sigma$  is large, the optimization problem is known to be numerically stiff. Note that the minimizer  $\hat{\theta}_{k+1}^*$  of the  $\sigma^{(k+1)}$ -problem (i.e., minimizing  $\hat{\mathcal{L}}_{\sigma^{(k+1)}}(w_\theta, \hat{\gamma}^*)$ ) can be initialized to  $\hat{\theta}_k^*$  of the  $\sigma^{(k)}$ -problem to warm start the training process. Hence, for each fixed  $\sigma^{(k)}$  (except  $\sigma^{(1)}$ ), the initial parameter configuration is close to the optimal one, and the training only requires a few iterations to reach convergence. The overall procedure is shown in Algorithm 3.1 for 2D problems with corner singularities and/or mixed boundary conditions. The stopping condition at line 3 of the algorithm reads: the iteration terminates whenever the maximum parameter vector  $\sigma^*$  is reached or the validating error falls below a certain threshold.

PINN exhibits the so-called frequency principle during the training: the low-frequency components are learned at a faster rate than the high-frequency ones [63]. In this vein, SEPINN harnesses this fact: by extracting the singular part, the regular part is smoother and thus by the frequency bias is easier to learn.

*Remark 3.1.* In this work we have focused on enforcing boundary conditions with the penalty method, due to its ease of implementation. Yet the penalty approach incurs the consistency error and restricts the norm in which the error can be bounded. In practice one may also enforce the boundary conditions exactly using (approximate) distance functions for domains with simple geometry [62]. However, the construction of distance functions for a general polygonal domain is fairly complicated, and thus we do not pursue the idea in this work.

**3.2. Three-dimensional problem.** Now we develop SEPINN for the 3D Poisson problem with edge singularities.

**3.2.1. Singular function representation.** There are several different types of geometric singularities in the 3D case, and each case has to be dealt with separately [19]. We only study edge singularities, which cause strong solution singularities. Indeed, the  $H^2(\Omega)$ -regularity of the solution  $u$  of problem (1.1) is not affected by the presence of conic points [40, 30, 42]. Now we state the precise setting.

TABLE 3

The orthogonal basis  $\{Z_{j,n}(z)\}_{n=0}^{\infty}$  of  $L^2(0,l)$ , for the  $j$ th wedge, with different boundary conditions on the surfaces  $\Gamma_{z_1}$  and  $\Gamma_{z_2}$ .

$\Gamma_{z_2} \setminus \Gamma_{z_1}$	Dirichlet	Neumann
Dirichlet	$Z_{j,n}(z) = \sin(\xi_{j,n}z),$ $\xi_{j,n} = \frac{n\pi}{l}, n \in \mathbb{N} \cup \{0\}$	$Z_{j,0}(z) = 0, Z_{j,n}(z) = \cos(\xi_{j,n}z),$ $\xi_{j,n} = (n - \frac{1}{2})\frac{\pi}{l}, n \in \mathbb{N}$
Neumann	$Z_{j,0}(z) = 0, Z_{j,n}(z) = \sin(\xi_{j,n}z),$ $\xi_{j,n} = (n - \frac{1}{2})\frac{\pi}{l}, n \in \mathbb{N}$	$Z_{j,n}(z) = \cos(\xi_{j,n}z),$ $\xi_{j,n} = \frac{n\pi}{l}, n \in \mathbb{N} \cup \{0\}$

Let  $\Omega_0 \subset \mathbb{R}^2$  be a polygonal domain as in section 3.1, and  $\Omega = \Omega_0 \times (0, l)$ . Like before, let  $\mathbf{v}_j, j = 1, 2, \dots, M$ , be the vertices of  $\Omega_0$  whose interior angles  $\omega_j, j = 1, 2, \dots, M$ , satisfy (3.3), and let  $\Gamma_{z_1} = \Omega_0 \times \{0\}$  and  $\Gamma_{z_2} = \Omega_0 \times \{l\}$ . We employ a complete orthogonal system  $\{Z_{j,n}\}_{n=0}^{\infty}$  of  $L^2(0, l)$ , given in Table 3.

Now we assume that near each edge  $\mathbf{v}_j \times (0, l)$ , the domain  $\Omega$  coincides with a 3D wedge  $G_j = \{(x_j, y_j, z) \in \Omega : 0 < r_j < R_j, 0 < \theta_j < \omega_j, 0 < z < l\}$ , where  $(r_j, \theta_j)$  are local polar coordinates (of the local Cartesian coordinates  $(x_j, y_j)$ ); cf. Figure 1(b) for a sketch. The functions  $u \in L^2(\Omega)$  and  $f \in L^2(\Omega)$  from problem (1.1) can then be represented by the following convergent Fourier series in the 3D wedge  $G_j$  (by suppressing the subscript  $j$ ):

$$(3.11) \quad u(x, y, z) = \frac{1}{2}u_0(x, y)Z_0(z) + \sum_{n=1}^{\infty} u_n(x, y)Z_n(z),$$

$$(3.12) \quad f(x, y, z) = \frac{1}{2}f_0(x, y)Z_0(z) + \sum_{n=1}^{\infty} f_n(x, y)Z_n(z),$$

where  $\{u_n\}_{n \in \mathbb{N}}$  and  $\{f_n\}_{n \in \mathbb{N}}$  are defined on the 2D domain  $\Omega_0$  by

$$u_n(x, y) = \frac{2}{l} \int_0^l u(x, y, z)Z_n(z)dz \quad \text{and} \quad f_n(x, y) = \frac{2}{l} \int_0^l f(x, y, z)Z_n(z)dz.$$

Substituting (3.11) and (3.12) into (1.1) gives countably many 2D elliptic problems:

$$(3.13) \quad \begin{cases} -\Delta u_n + \xi_{j,n}^2 u_n = f_n & \text{in } \Omega_0, \\ u_n = 0 & \text{on } \Gamma_D, \\ \partial_n u_n = 0 & \text{on } \Gamma_N. \end{cases}$$

Then problem (1.1) can be analyzed via the 2D problems. Below we describe the edge behavior of the weak solution  $u \in H^1(\Omega)$  [55, Theorem 2.1]. The next theorem gives a crucial decomposition of  $u \in H^{1+\frac{\pi}{\omega^*}-\epsilon}$  for every  $\epsilon > 0$ . The functions  $\Psi_{j,i}$  are the so-called edge flux intensity functions.

**THEOREM 3.1.** *For any fixed  $f \in L^2(\Omega)$ , let  $u \in H^1(\Omega)$  be the unique weak solution to problem (1.1). Then there exist unique functions  $\Psi_{j,i} \in H^{1-\lambda_{j,i}}(0, l)$  of the variable  $z$  such that  $u$  can be split into a sum of a regular part  $w \in H^2(\Omega)$  and a singular part  $S$  with the following properties:*

$$(3.14a) \quad u = w + S, \quad S = \sum_{j=1}^M \sum_{i \in \mathbb{I}_j} S_{j,i}(x_j, y_j, z),$$

$$(3.14b) \quad S_{j,i}(x_j, y_j, z) = (T_j(r_j, z) * \Psi_{j,i}(z))\eta_{\rho_j}(r_j)s_{j,i}(r_j, \theta_j),$$

where the functions  $T_j$  are fixed Poisson's kernels, and the symbol  $*$  denotes the convolution in  $z \in (0, l)$ . Moreover, there exists a constant  $C > 0$  independent of  $f \in L^2(\Omega)$  such that  $\|w\|_{H^2(\Omega)} \leq C\|f\|_{L^2(\Omega)}$ .

The explicit form of the function  $T_j(r_j, z) * \Psi_{j,i}(z)$  and the formula for the coefficients  $\gamma_{j,i,n}$  are given below [55, Theorem 2.2], [54, pp. 179–182].

**THEOREM 3.2.** *The coefficients  $\Phi_{j,i}(x_j, y_j, z) = T_j(r_j, z) * \Psi_{j,i}(z)$  of the singularities in (3.14b) can be represented by Fourier series in  $z$  and with respect to the orthogonal system  $\{Z_n(z)\}_{n=0}^\infty$ :*

$$\begin{aligned}
 T_j(r_j, z) &= \frac{1}{2}Z_0(z) + \sum_{n=1}^\infty e^{-\xi_{j,n}r_j} Z_n(z), \\
 \Psi_{j,i}(z) &= \frac{1}{2}\gamma_{j,i,0}Z_0(z) + \sum_{n=1}^\infty \gamma_{j,i,n}Z_n(z), \\
 \Phi_{j,i}(x_j, y_j, z) &= \frac{1}{2}\gamma_{j,i,0}Z_0(z) + \sum_{n=1}^\infty \gamma_{j,i,n}e^{-\xi_{j,n}r_j} Z_n(z),
 \end{aligned}
 \tag{3.15}$$

where  $\gamma_{j,i,n}$  are given by  $\gamma_{j,i,n} = \frac{2}{l\omega_j\lambda_{j,i}} \int_{G_j} f_j^* e^{\xi_{j,n}r_j} s_{j,-k}(r_j, \theta_j) Z_n(z) dx dy dz$ , and  $f_j^* = f\eta_{\rho_j} - u(\frac{\partial^2 \eta_{\rho_j}}{\partial r_j^2} + (2\xi_{j,n} + \frac{1}{r_j})\frac{\partial \eta_{\rho_j}}{\partial r_j} + (2\xi_{j,n}^2 + \frac{\xi_{j,n}}{r_j})\eta_{\rho_j}) - 2\frac{\partial u}{\partial r_j}(\frac{\partial \eta_{\rho_j}}{\partial r_j} + \xi_{j,n}\eta_{\rho_j})$ . Moreover there exists a constant  $C > 0$  independent of  $f$  such that

$$|\gamma_{j,i,0}|^2 + \sum_{n=1}^\infty \xi_{j,n}^{2(1-\lambda_{j,i})} |\gamma_{j,i,n}|^2 \leq C\|f\|_{L^2(G_j)}.
 \tag{3.16}$$

Note that the formula for  $\gamma_{j,i,n}$  involves a singular integral and is numerically inconvenient to evaluate. We discuss the case of only one edge with one singular function below. Then the solution  $u$  can be split into

$$u = w + \Phi\eta_\rho s.
 \tag{3.17}$$

Since the functions  $\Phi\eta_\rho s$  and  $\partial_n(\Phi\eta_\rho s)$  vanish on  $\Gamma_D$  and  $\Gamma_N$ , respectively,  $w$  solves

$$\begin{cases} -\Delta w = f + \Delta(\Phi\eta_\rho s) & \text{in } \Omega, \\ w = 0 & \text{on } \Gamma_D, \\ \partial_n w = 0 & \text{on } \Gamma_N. \end{cases}
 \tag{3.18}$$

This forms the basis of SEPINN for 3D problems with edge singularities. Below we describe two strategies for constructing SEPINN approximations, i.e., SEPINN-C based on a cutoff approximation of the infinite series and SEPINN-N based on multiple DNN approximations.

**3.2.2. SEPINN–cutoff approximation.** The expansion (3.15) of  $\Phi$  in (3.18) involves infinitely many unknown scalar coefficients  $\{\gamma_{j,i,n}\}_{n=0}^\infty$ . In practice, it is infeasible to learn all of them. However, since the series is convergent, we may truncate it to a finite number of terms: the edge flux intensity function  $\Phi_{j,i}(x_j, y_j, z)$  from (3.15) is approximated by the truncation

$$\Phi_{j,i}^N(x_j, y_j, z) = \frac{1}{2}\gamma_{j,i,0}Z_0(z) + \sum_{n=1}^N \gamma_{j,i,n}e^{-\xi_{j,n}r_j} Z_n(z) \quad \text{with } N \in \mathbb{N}.
 \tag{3.19}$$

The approximate singular function  $S_{j,i}^N$  is given by

$$(3.20) \quad S_{j,i}^N(x_j, y_j, z) = \Phi_{j,i}^N(x_j, y_j, z) \eta_{\rho_j}(r_j) s_{j,i}(r_j, \theta_j).$$

In view of the splitting (3.14a) and truncation (3.20), the approximate regular part  $w$  satisfies with  $\tilde{f} = f + \gamma_0 \Delta(\eta_\rho s) + \sum_{n=1}^N \gamma_n \Delta(e^{-\xi_n r} Z_n(z) \eta_\rho s)$ ,

$$(3.21) \quad \begin{cases} -\Delta w = \tilde{f} & \text{in } \Omega, \\ w = 0 & \text{on } \Gamma_D, \\ \partial_n w = 0 & \text{on } \Gamma_N, \end{cases}$$

with  $\gamma_i, i = 0, 1, \dots, N$ , being  $N + 1$  unknown parameters. Let  $w^N$  be the solution of (3.21) and  $w^*$  the solution of (3.18). The next result shows that when  $N$  is large enough, the error  $w^N - w^*$  can be made small, as is the error between  $u^N$  and  $u^*$ , which underpins the truncation method.

**THEOREM 3.3.** *Let  $w^N$  be the solution of (3.21) and  $S^N$  be the truncated singular function. Let  $u^N = w^N + S^N$  and let  $u^*$  be the solution of (1.1). Then there holds*

$$(3.22) \quad \|u^N - u^*\|_{H^1(\Omega)} \leq CN^{-1} \|f\|_{L^2(G)}.$$

*Proof.* By [55, Lemma 3.3], there exists a constant  $C > 0$  independent of  $f$  such that

$$(3.23) \quad \|S^N - S^*\|_{H^1(G)} \leq CN^{-1} \|f\|_{L^2(G)}.$$

Since  $w^N$  solves (3.21) and  $w^*$  solves (3.18),  $w^* - w^N$  satisfies problem (1.1) with the source  $f^N = \sum_{n=N+1}^\infty \gamma_n \Delta(e^{-\xi_n r} Z_n(z) \eta_\rho s)$ . By elliptic regularity theory, we have

$$(3.24) \quad \|w^* - w^N\|_{H^1(\Omega)} \leq C \|f^N\|_{H_{\Gamma_D}^{-1}(\Omega)} \leq C \|\Delta(S^N - S^*)\|_{H_{\Gamma_D}^{-1}(\Omega)},$$

where the notation  $H_{\Gamma_D}^{-1}(\Omega)$  denotes the dual space of  $H_{\Gamma_D}^1(\Omega) = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$ . Upon integration by parts, we get

$$\begin{aligned} \|\Delta(S^N - S^*)\|_{H_{\Gamma_D}^{-1}(\Omega)} &= \sup_{v \in H_{\Gamma_D}^1(\Omega), \|v\|_{H^1(\Omega)} \leq 1} \left| \int_{\Omega} \Delta(S^N - S^*) v \, dx \right| \\ &= \sup_{v \in H_{\Gamma_D}^1(\Omega), \|v\|_{H^1(\Omega)} \leq 1} \left| \int_{\Omega} \nabla(S^N - S^*) \cdot \nabla v \, dx \right| \leq \|S^N - S^*\|_{H^1(\Omega)} \\ &= \|S^N - S^*\|_{H^1(G)}. \end{aligned}$$

Combining this estimate with (3.23) yields the desired assertion.  $\square$

Following section 2.2, we approximate  $w^N$  by an element  $w_\theta \in \mathcal{A}$ . Like in the 2D case, we view the parameters  $\gamma_N := (\gamma_0, \dots, \gamma_N)$  as trainable parameters and learn them along with DNN parameters  $\theta$ . Thus, we employ the empirical loss

$$\hat{\mathcal{L}}_\sigma(w_\theta; \gamma_N) = \frac{|\Omega|}{N_r} \sum_{i=1}^{N_r} (\Delta w_\theta(X_i) + \tilde{f}(X_i))^2 + \sigma_d \frac{|\Gamma_D|}{N_d} \sum_{j=1}^{N_d} w_\theta^2(Y_j) + \sigma_n \frac{|\Gamma_N|}{N_n} \sum_{k=1}^{N_n} (\partial_n w_\theta(Z_k))^2$$

with i.i.d. sampling points  $\{X_i\}_{i=1}^{N_r} \sim U(\Omega)$ ,  $\{Y_j\}_{j=1}^{N_d} \sim U(\Gamma_D)$ , and  $\{Z_k\}_{k=1}^{N_n} \sim U(\Gamma_N)$ . The empirical loss  $\hat{\mathcal{L}}_\sigma(w_\theta; \gamma_N)$  can be minimized using the two-stage procedure as in the 2D case.

**3.2.3. SEPINN—neural networks approximation.** There is actually a direct way to resolve the singular part  $S$ , i.e., using a DNN to approximate  $\Phi$  in (3.14a). We term the resulting method SEPINN-N. This strategy eliminates the necessity of explicitly knowing the expansion basis and relieves us from lengthy derivations. Thus, compared with SEPINN-C, it is more direct and simpler to implement. The downside is an increase in the number of parameters that need to be learned. Specifically, let  $\mathcal{B}$  be a DNN function class with a fixed architecture (possibly different from  $\mathcal{A}$ ) and  $\zeta$  its parameterization. Then a DNN  $\Phi_\zeta \in \mathcal{B}$  is employed to approximate  $\Phi$  in (3.17), where the DNN parameters  $\zeta$  are also learned. The splitting is then given by

$$(3.25) \quad u = w + \Phi_\zeta \eta_\rho s.$$

Since we cannot guarantee  $\Phi_\zeta = 0$  on  $\Gamma_D$  or  $\partial_n \Phi_\zeta = 0$  on  $\Gamma_N$ , the boundary conditions of  $w$  have to be modified accordingly (noting  $\partial_n(\eta_\rho s) = 0$ ):

$$(3.26) \quad \begin{cases} -\Delta w = f + \Delta(\Phi_\zeta \eta_\rho s) & \text{in } \Omega, \\ w = -\Phi_\zeta \eta_\rho s & \text{on } \Gamma_D, \\ \partial_n w = -\partial_n(\Phi_\zeta) \eta_\rho s & \text{on } \Gamma_N. \end{cases}$$

Like before, we can obtain the empirical loss

$$(3.27) \quad \begin{aligned} \widehat{\mathcal{L}}_\sigma(w_\theta; \Phi_\zeta) &= \frac{|\Omega|}{N_r} \sum_{i=1}^{N_r} (\Delta w_\theta(X_i) + f(X_i) + \Delta(\Phi_\zeta \eta_\rho s)(X_i))^2 \\ &+ \sigma_d \frac{|\Gamma_D|}{N_d} \sum_{j=1}^{N_d} (w_\theta(Y_j) + \Phi_\zeta \eta_\rho s(Y_j))^2 + \sigma_n \frac{|\Gamma_N|}{N_n} \sum_{k=1}^{N_n} (\partial_n w_\theta(Z_k) \\ &+ \partial_n(\Phi_\zeta) \eta_\rho s(Z_k))^2 \end{aligned}$$

with i.i.d. sampling points  $\{X_i\}_{i=1}^{N_r} \sim U(\Omega)$ ,  $\{Y_j\}_{j=1}^{N_d} \sim U(\Gamma_D)$ , and  $\{Z_k\}_{k=1}^{N_n} \sim U(\Gamma_N)$ . The implementation of SEPINN-N is direct, since both DNNs  $w_\theta$  and  $\Phi_\zeta$  are learned, and the resulting optimization problem can be minimized directly.

**4. Error analysis.** Now we discuss the error analysis of SEPINN developed in section 3, following the strategies established in the recent works [34, 48, 32], in order to provide theoretical guarantee of SEPINN. We only analyze the 2D problem in section 3.1. Let  $u^* = w^* + \gamma^* \eta_\rho s$  be the exact solution to problem (1.1), in which case  $(w^*, \gamma^*)$  is also a global minimizer of the loss  $\mathcal{L}_\sigma(w, \gamma)$  (with a zero loss value). Moreover, we assume that  $w^* \in H^3(\Omega)$  and  $|\gamma^*| \leq B_\gamma$ ; cf. (3.7). The  $H^3(\Omega)$  regularity of the smooth part  $w^*$  is needed for controlling the error of approximating  $w$  in the  $H^2(\Omega)$  norm using DNNs. If the assumption does not hold, we can split out additional singular functions; see the argument in section 3.1.1. Several preliminary results used in the analysis are given in the appendix.

The following approximation property holds [31, Proposition 4.8]. The notation  $(s = 2)$  equals to 1 if  $s = 2$  and zero otherwise.

**LEMMA 4.1.** *Let  $s \in \mathbb{N} \cup \{0\}$  and  $p \in [1, \infty]$  be fixed, and  $v \in W^{k,p}(\Omega)$  with  $k \geq s + 1$ . Then for any tolerance  $\epsilon > 0$ , there exists at least one  $v_\theta$  of depth  $\mathcal{O}(\log(d + k))$ , with  $|\theta|_{\ell^0}$  bounded by  $\mathcal{O}(\epsilon^{-\frac{d}{k-s-\mu(s=2)}})$  and  $|\theta|_{\ell^\infty}$  by  $\mathcal{O}(\epsilon^{-2 - \frac{2(d/p+d+s+\mu(s=2))+d/p+d}{k-s-\mu(s=2)}})$ , where  $\mu > 0$  is arbitrarily small, such that*

$$\|v - v_\theta\|_{W^{s,p}(\Omega)} \leq \epsilon.$$

For  $\epsilon > 0$ , with  $d = 2$ ,  $k = 3$ , and  $s = 2$ , by Lemma 4.1, there exists a DNN

$$(4.1) \quad v_\theta \in \mathcal{N}_\rho(C, C\epsilon^{-\frac{2}{1-\mu}}, C\epsilon^{-2-\frac{16+2\mu}{1-\mu}}) =: \mathcal{W}_\epsilon$$

such that  $\|w^* - v_\theta\|_{H^2(\Omega)} \leq \epsilon$ . Also let  $I_\gamma = [-B_\gamma, B_\gamma]$ . Let  $(\widehat{w}_\theta, \widehat{\gamma})$  be a minimizer of  $\widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  (cf. (3.10)) over  $\mathcal{W}_\epsilon \times I_\gamma$ , and set  $\widehat{u} = \widehat{w}_\theta + \widehat{\gamma}\eta_\rho s$ . The next lemma gives a decomposition of the error  $\|u^* - \widehat{u}\|_{L^2(\Omega)}$ .

LEMMA 4.2. *For any  $\epsilon > 0$ , let  $(\widehat{w}_\theta, \widehat{\gamma}) \in \mathcal{W}_\epsilon \times I_\gamma$  be a minimizer to the loss  $\widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)$ . Then there holds*

$$\begin{aligned} \|u^* - \widehat{u}\|_{L^2(\Omega)}^2 &\leq c \min(\sigma_d, \sigma_n)^{-1} \mathcal{L}_\sigma(\widehat{w}_\theta, \widehat{\gamma}) \\ &\leq c \min(\sigma_d, \sigma_n)^{-1} \left( (1 + \sigma_d + \sigma_n)\epsilon^2 + \sup_{(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma} |\mathcal{L}_\sigma(w_\theta, \gamma) - \widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)| \right), \end{aligned}$$

where the constant  $c$  is independent of  $\sigma$ .

*Proof.* For any  $(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma$ , let  $u = w_\theta + \gamma\eta_\rho s$ , and denote the corresponding overall error by  $e = u^* - u$ . By the trace theorem, we have

$$\mathcal{L}_\sigma(w_\theta, \gamma) = \|\Delta e\|_{L^2(\Omega)}^2 + \sigma_d \|e\|_{L^2(\Gamma_D)}^2 + \sigma_n \|\partial_n e\|_{L^2(\Gamma_N)}^2 \leq c(1 + \sigma_d + \sigma_n) \|e\|_{H^2(\Omega)}^2.$$

To treat the nonzero boundary conditions of  $w_\theta$ , we define the extension  $\zeta$  by

$$\begin{cases} -\Delta \zeta = 0 & \text{in } \Omega, \\ \zeta = w_\theta & \text{on } \Gamma_D, \\ \partial_n \zeta = \partial_n w_\theta & \text{on } \Gamma_N. \end{cases}$$

Then the following elliptic regularity estimate holds [9, Theorem 4.2, p. 870]:

$$(4.2) \quad \|\zeta\|_{L^2(\Omega)} \leq c(\|w_\theta\|_{L^2(\Gamma_D)} + \|\partial_n w_\theta\|_{L^2(\Gamma_N)}).$$

This estimate bounds the consistency error due to the boundary penalty. Let  $\tilde{e} = e + \zeta$ . Then it satisfies

$$\begin{cases} -\Delta \tilde{e} = -\Delta e & \text{in } \Omega, \\ \tilde{e} = 0 & \text{on } \Gamma_D, \\ \partial_n \tilde{e} = 0 & \text{on } \Gamma_N. \end{cases}$$

Since  $\Delta e \in L^2(\Omega)$ , the Poincaré inequality and the standard energy argument imply

$$\|\tilde{e}\|_{L^2(\Omega)} \leq c\|\nabla \tilde{e}\|_{L^2(\Omega)} \leq c\|\Delta e\|_{L^2(\Omega)}.$$

This, the stability estimate (4.2) and the triangle inequality lead to

$$\|e\|_{L^2(\Omega)}^2 \leq c(\|\tilde{e}\|_{L^2(\Omega)}^2 + \|\zeta\|_{L^2(\Omega)}^2) \leq c \min(\sigma_d, \sigma_n)^{-1} \mathcal{L}_\sigma(w_\theta, \gamma).$$

This proves the first inequality of the lemma. Next, by Lemma 4.1 and the assumption  $w^* \in H^3(\Omega)$ , there exists  $w_{\bar{\theta}} \in \mathcal{W}_\epsilon$  such that  $\|w_{\bar{\theta}} - w^*\|_{H^2(\Omega)} \leq \epsilon$ . Let  $\bar{u} = w_{\bar{\theta}} + \gamma^*\eta_\rho s$ . Then we derive

$$\begin{aligned} \mathcal{L}_\sigma(w_{\bar{\theta}}, \gamma^*) &= \|\Delta(w_{\bar{\theta}} - w^*)\|_{L^2(\Omega)}^2 + \sigma_d \|w_{\bar{\theta}} - w^*\|_{L^2(\Gamma_D)}^2 + \sigma_n \|\partial_n(w_{\bar{\theta}} - w^*)\|_{L^2(\Gamma_N)}^2 \\ &\leq c(1 + \sigma_d + \sigma_n) \|w_{\bar{\theta}} - w^*\|_{H^2(\Omega)}^2 \leq c(1 + \sigma_d + \sigma_n)\epsilon^2. \end{aligned}$$

Thus by the minimizing property of  $(\widehat{w}_\theta, \widehat{\gamma})$ , we arrive at

$$\begin{aligned} \mathcal{L}_\sigma(\widehat{w}_\theta, \widehat{\gamma}) &\leq |\mathcal{L}_\sigma(\widehat{w}_\theta, \widehat{\gamma}) - \widehat{\mathcal{L}}_\sigma(\widehat{w}_\theta, \widehat{\gamma})| + |\widehat{\mathcal{L}}_\sigma(w_{\bar{\theta}}, \gamma^*) - \mathcal{L}_\sigma(w_{\bar{\theta}}, \gamma^*)| + \mathcal{L}_\sigma(w_{\bar{\theta}}, \gamma^*) \\ &\leq 2 \sup_{(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma} |\mathcal{L}_\sigma(w_\theta, \gamma) - \widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)| + c(1 + \sigma_d + \sigma_n)\epsilon^2. \end{aligned}$$

This completes the proof of the lemma. □

*Remark 4.1.* The error estimate in Lemma 4.2 is given in terms of the  $L^2(\Omega)$  norm. This is due to the use of the penalty method for treating the zero Dirichlet boundary condition; cf. (4.2). This is a form of the consistency error, and it essentially prevents one from obtaining error estimates in the  $H^1(\Omega)$  norm. The estimate indicates that one should take the parameter  $\sigma$  sufficiently large in order to have small consistency errors. See also the works [35, 53, 32] for detailed discussions on the consistency error due to penalization for the Dirichlet boundary condition in the context of the DRM.

Next, we bound the error  $\mathcal{E}_{stat} = \sup_{(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma} |\mathcal{L}_\sigma(w_\theta, \gamma) - \widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)|$ , which arises from approximating the integrals by Monte Carlo. By the triangle inequality, we have the splitting

$$\begin{aligned} \mathcal{E}_{stat} &\leq \sup_{(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma} |\Omega| \left| \frac{1}{N_r} \sum_{i=1}^{N_r} h_r(X_i; w_\theta, \gamma) - \mathbb{E}_X(h_r(X; w_\theta, \gamma)) \right| \\ (4.3) \quad &+ \sup_{w_\theta \in \mathcal{W}_\epsilon} \sigma_d |\Gamma_D| \left| \frac{1}{N_d} \sum_{j=1}^{N_d} h_d(Y_j; w_\theta) - \mathbb{E}_Y(h_d(Y; w_\theta)) \right| \\ &+ \sup_{w_\theta \in \mathcal{W}_\epsilon} \sigma_n |\Gamma_N| \left| \frac{1}{N_n} \sum_{k=1}^{N_n} h_n(Z_k; w_\theta) - \mathbb{E}_Y(h_n(Z; w_\theta)) \right| \end{aligned}$$

with  $h_r(x; w_\theta, \gamma) = (\Delta w_\theta + f + \gamma \Delta(\eta_\rho s))^2(x)$  for  $x \in \Omega$ ,  $h_d(y; w_\theta) = |w_\theta(y)|^2$  for  $y \in \Gamma_D$ , and  $h_n(z; w_\theta) = |\partial_n w_\theta(z)|^2$  for  $z \in \Gamma_N$ . Thus, we define the following three function classes:  $\mathcal{H}_r = \{h_r(w_\theta, \gamma) : w_\theta \in \mathcal{W}_\epsilon, \gamma \in I_\gamma\}$ ,  $\mathcal{H}_d = \{h_d(w_\theta) : w_\theta \in \mathcal{W}_\epsilon\}$ , and  $\mathcal{H}_n = \{h_n(w_\theta) : w_\theta \in \mathcal{W}_\epsilon\}$ .

To bound the error components in the decomposition (4.3), we employ Rademacher complexity of the DNN function classes  $\mathcal{H}_r$ ,  $\mathcal{H}_d$ , and  $\mathcal{H}_n$ , which is then bounded using Dudley’s formula in Lemma A.6 and Lipschitz continuity of the functions in these DNN function classes in Lemma A.4. These technical details are given in the appendix. Then we can state the following bound on the quadrature error.

LEMMA 4.3. *For any small  $\tau$ , with probability at least  $1 - 3\tau$ , there holds*

$$\sup_{(w_\theta, \gamma) \in \mathcal{W}_\epsilon \times I_\gamma} |\mathcal{L}_\sigma(w_\theta, \gamma) - \widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)| \leq c(e_r + \sigma_d e_d + \sigma_n e_n)$$

with  $c = c(\|f\|_{L^\infty(\Omega)}, \|\Delta(\eta_\rho s)\|_{L^\infty(\Omega)})$ , and  $e_r$ ,  $e_d$ , and  $e_n$  defined by

$$\begin{aligned} e_r &\leq c \frac{L^2 B_\theta^{4L} N_\theta^{4L-4} (N_\theta^{\frac{1}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} N_r) + \log^{\frac{1}{2}} \frac{1}{\tau})}{\sqrt{N_r}}, \\ e_d &\leq c \frac{B_\theta^2 N_\theta^2 (N_\theta^{\frac{1}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} N_d) + \log^{\frac{1}{2}} \frac{1}{\tau})}{\sqrt{N_d}}, \\ e_n &\leq c \frac{B_\theta^{2L} N_\theta^{2L-2} (N_\theta^{\frac{1}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} N_n) + \log^{\frac{1}{2}} \frac{1}{\tau})}{\sqrt{N_n}}. \end{aligned}$$



*Proof.* Fix  $m \in \mathbb{N}$ ,  $B_\theta \in [1, \infty)$ ,  $\epsilon \in (0, 1)$ , and  $\mathbb{B}_{B_\theta} := \{x \in \mathbb{R}^m : |x|_{\ell^\infty} \leq B_\theta\}$ . Then by [16, Proposition 5],  $\log \mathcal{C}(\mathbb{B}_{B_\theta}, |\cdot|_{\ell^\infty}, \epsilon) \leq m \log(4B_\theta \epsilon^{-1})$ . The Lipschitz continuity estimates in Lemmas A.3 and A.4 imply

$$\log \mathcal{C}(\mathcal{H}_r, \|\cdot\|_{L^\infty(\Omega)}, \epsilon) \leq \log \mathcal{C}(\Theta \times I_\gamma, \max(|\cdot|_{\ell^\infty}, |\cdot|), \Lambda_r^{-1} \epsilon) \leq cN_\theta \log(4B_\theta \Lambda_r \epsilon^{-1})$$

with  $\Lambda_r = cN_\theta L^3 W^{5L-5} B_\theta^{5L-3}$ . By Lemma A.4, we have  $M_{\mathcal{H}_r} = cL^2 W^{4L-4} B_\theta^{4L}$ . Then setting  $s = n^{-\frac{1}{2}}$  in Lemma A.6 and using the facts  $1 \leq B_\theta$ ,  $1 \leq L$ , and  $1 \leq W \leq N_\theta$ ,  $1 \leq L \leq c \log 5$  (for  $d=2$  and  $k=3$ ) (cf. Lemma 4.1) lead to

$$\begin{aligned} \mathfrak{R}_n(\mathcal{H}_r) &\leq 4n^{-\frac{1}{2}} + 12n^{-\frac{1}{2}} \int_{n^{-\frac{1}{2}}}^{M_{\mathcal{H}_r}} (cN_\theta \log(4B_\theta \Lambda_r \epsilon^{-1}))^{\frac{1}{2}} d\epsilon \\ &\leq 4n^{-\frac{1}{2}} + 12n^{-\frac{1}{2}} M_{\mathcal{H}_r} (cN_\theta \log(4B_\theta \Lambda_r n^{\frac{1}{2}}))^{\frac{1}{2}} \\ &\leq 4n^{-\frac{1}{2}} + cn^{-\frac{1}{2}} W^{4L-4} B_\theta^{4L} N_\theta^{\frac{1}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} \Lambda_r + \log^{\frac{1}{2}} n) \\ &\leq cn^{-\frac{1}{2}} B_\theta^{4L} N_\theta^{4L-\frac{7}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} n). \end{aligned}$$

Similarly, repeating the preceding argument leads to

$$\begin{aligned} \mathfrak{R}_n(\mathcal{H}_d) &\leq cn^{-\frac{1}{2}} B_\theta^2 N_\theta^{\frac{5}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} n), \\ \mathfrak{R}_n(\mathcal{H}_n) &\leq cn^{-\frac{1}{2}} B_\theta^{2L} N_\theta^{2L-\frac{3}{2}} (\log^{\frac{1}{2}} B_\theta + \log^{\frac{1}{2}} N_\theta + \log^{\frac{1}{2}} n). \end{aligned}$$

Finally, the desired result follows from Lemma A.2.  $\square$

Then combining Lemma 4.2 with Lemma 4.3 yields the following error estimate. Thus, by choosing the numbers  $N_r$ ,  $N_d$ , and  $N_n$  of sampling points sufficiently large, the  $L^2(\Omega)$  error of the SEPINN approximation can be made about  $O(\epsilon^2)$ .

**THEOREM 4.4.** *Fix a tolerance  $\epsilon > 0$ , and let  $(\hat{w}_\theta, \hat{\gamma}) \in \mathcal{W}_\epsilon \times I_\gamma$  be a minimizer to the empirical loss  $\hat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  in (3.10). Then for any small  $\tau$ , with the statistical errors  $e_r$ ,  $e_d$ , and  $e_n$  from Lemma 4.3, we have with probability at least  $1 - 3\tau$ , the following error estimate holds:*

$$\|u^* - \hat{u}\|_{L^2(\Omega)}^2 \leq c \min(\sigma_d, \sigma_n)^{-1} ((1 + \sigma_d + \sigma_n) \epsilon^2 + e_r + \sigma_d e_d + \sigma_n e_n),$$

where the constant  $c$  is independent of  $\sigma$ .

*Remark 4.2.* According to Theorem 4.4, the parameter  $\sigma$  has to be tuned carefully in order to optimally achieve the error: a too small  $\sigma$  incurs large consistency errors, while a too large  $\sigma$  incurs big statistical errors (on approximating the integrals on the boundary).

**5. Numerical experiments.** Now we present numerical examples to illustrate SEPINN and compare it with several existing PINN type solvers. Note that our main goal is to illustrate the improved approximation accuracy by singularity enrichment, instead of verifying the convergence rate. Indeed, even for the standard PINN (or any other neural solvers), numerically verifying the theoretical rate remains a daunting challenge [59]. In the training,  $N_r = 10,000$  points in the domain  $\Omega$  and  $N_b = 800$  points on the boundary  $\partial\Omega$  are selected uniformly at random to form the empirical loss  $\hat{\mathcal{L}}_\sigma$ , unless otherwise specified. In the PF strategy, we take an increasing factor  $q = 1.5$ . All numerical experiments were conducted on a personal laptop (Windows 10, with RAM 8.0 GB, Intel Core i7-10510U CPU, 2.3 GHz), with Python 3.9.7,

with PyTorch. The gradient of the DNN output  $w_\theta(x)$  with respect to the input  $x$  (i.e., spatial derivative) and that of the loss  $\widehat{\mathcal{L}}_\sigma$  to  $\theta$  are computed via automatic differentiation [7] using `torch.autograd`.

For SEPINN and SEPINN-C (based on cutoff), we minimize the loss  $\widehat{\mathcal{L}}_\sigma$  in two stages: first determine the coefficients  $\gamma_N$ , and then reduce the boundary error and refine the DNN approximation  $w_\theta$  (of the regular part  $w$ ) by the PF strategy. We use different optimizers at these two stages. First, we minimize the loss  $\widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  in both  $\theta$  and  $\gamma$  using Adam [39] (from the SciPy library), with the default setting (tolerance: 1.0e-8; no box constraint; maximum iteration number: 1000); then, we minimize  $\widehat{\mathcal{L}}_\sigma(w_\theta, \widehat{\gamma}^*)$  (fixing  $\gamma$  at  $\widehat{\gamma}^*$  fixed) using limited memory BFGS (L-BFGS) [11], with the default setting (tolerance: 1.0e-9; no box constraint; maximum iteration number: 2500). For SEPINN-N, we employ only L-BFGS [11]. We have employed different optimizers for SEPINN-C and SEPINN-N: in SEPINN-C, the influence of the DNN parameters  $\theta$  and stress intensity factors  $\gamma$  on the loss  $\widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  differ markedly, and one may use different learning rates for them to compensate the influences. Once the parameter  $\gamma$  is fixed, the loss  $\widehat{\mathcal{L}}_\sigma(w_\theta, \gamma)$  can be efficiently minimized via L-BFGS. To measure the accuracy of an approximation  $\widehat{w}$  of  $w^*$ , we use the relative  $L^2(\Omega)$ -error  $e = \|w^* - \widehat{w}\|_{L^2(\Omega)} / \|w^*\|_{L^2(\Omega)}$ , with the error computed using sampling points in the domain  $\Omega$ . The stopping condition of the PF strategy is set to  $e < 1.00\text{e-}3$  and  $\sigma_d^{(k)}, \sigma_n^{(k)} \leq \sigma^*$ , for some fixed  $\sigma^* > 0$ . The first condition ensures that  $\widehat{w}$  can achieve the desired accuracy and the second condition terminates the iteration after a finite number of loops. The detailed hyperparameter setting of the PF strategy is listed in Table 4. Throughout the training the box constraint on the DNN parameter is not imposed, since numerically it is observed that the DNN parameters stay bounded during the entire training process. The Python code for reproducing the numerical experiments will be made available at <https://github.com/hhjc-web/SEPINN.git>. The zip file of the complete set of codes is available in the supplementary material (M160119\_SuppMat.pdf [local/web 399KB]).

First, we showcase the approach on an L-shaped domain [15, Example 5.2].

*Example 5.1.* The domain  $\Omega = (-1, 1)^2 \setminus ([0, 1] \times (-1, 0])$ . Set  $\rho = 1$  and  $R = \frac{1}{2}$  in (3.4), the source

$$f = \begin{cases} \sin(2\pi x) [2\pi^2 (y^2 + 2y) (y^2 - 1) - (6y^2 + 6y - 1)] - \Delta(\eta_\rho s), & -1 \leq y \leq 0, \\ \sin(2\pi x) [2\pi^2 (-y^2 + 2y) (y^2 - 1) - (-6y^2 + 6y + 1)] - \Delta(\eta_\rho s), & 0 \leq y \leq 1, \end{cases}$$

TABLE 4

The hyperparameters for the PF strategy for SEPINN (2D) and SEPINN-C and SEPINN-N (3D) for the examples. The notation  $\widehat{\gamma}^*$  denotes the estimated stress intensity factor, and  $e$  the prediction error.

Example	$\sigma^*$	$\sigma^{(1)}$	$\sigma^{(K)}$	$\widehat{\gamma}^*$	Epoch	Time(s)	$e$
5.1	1200	100	1139.1	1.0001	13.4k	690	1.84e-3
5.2	800	$\begin{pmatrix} 100 \\ 100 \end{pmatrix}$	$\begin{pmatrix} 759.4 \\ 759.4 \end{pmatrix}$	1.0033	3.7k	109	4.62e-3
5.3 (SE-C)	4000	400	3037.5	Table 5	8.3k	4161	5.04e-2
5.3 (SE-N)	5000	400	4556.3	--	13.2k	2095	3.80e-2
5.4 (SE-C)	1000	$\begin{pmatrix} 100 \\ 100 \end{pmatrix}$	$\begin{pmatrix} 759.4 \\ 759.4 \end{pmatrix}$	Fig. 8	10.4k	10071	2.08e-2
5.4 (SE-N)	4000	$\begin{pmatrix} 100 \\ 400 \end{pmatrix}$	$\begin{pmatrix} 759.4 \\ 3037.5 \end{pmatrix}$	--	21.2k	6493	3.83e-2
5.5	600	50	569.5	Table 7	15.8k	1396	--

with the singular function  $s = r^{\frac{2}{3}} \sin(\frac{2\theta}{3})$ , and  $\Gamma_D = \partial\Omega$ . The exact solution  $u$  of the problem is given by  $u = w + \eta_\rho s$ , with the regular part  $w$  given by

$$(5.1) \quad w = \begin{cases} \sin(2\pi x) (\frac{1}{2}y^2 + y) (y^2 - 1), & -1 \leq y \leq 0, \\ \sin(2\pi x) (-\frac{1}{2}y^2 + y) (y^2 - 1), & 0 \leq y \leq 1. \end{cases}$$

In this example, the regular part  $w$  belongs to  $H^2(\Omega)$  but not to  $H^3(\Omega)$ , and  $u$  lies in  $H^1(\Omega)$  but not in  $H^2(\Omega)$ . In SEPINN, we employ a 2-20-20-20-1 DNN (3 hidden layers, each having 20 neurons). The first stage of the PF strategy gives an estimate  $\hat{\gamma}^* = 1.0001$ , and the final prediction error  $e$  after the second stage is  $1.84e-3$ . Figure 2 shows that the pointwise error of the SEPINN approximation is small and the accuracy around the singularity at the reentrant corner is excellent. In contrast, applying PINN and DRM directly fails to yield satisfactory results near the reentrant corner because of the presence of the singular term  $r^{\frac{2}{3}} \sin \frac{2}{3}\theta$ , consistent with the approximation theory of DNNs to singular functions [31]. DRM shows larger errors over the whole domain, not just in the vicinity of the corner. By adaptively adjusting the empirical loss, SAPINN and FIPINN can yield more accurate approximations than PINN, but the error around the singularity is still large. Numerically, FIPINN can adaptively add sampling points near the corner but does not give high concentration, which limits the accuracy of the final DNN approximation. For all the methods, the largest error occurs near the boundary  $\partial\Omega$ .

To give further insights into the methods, we show in Figure 3 the training dynamics of the empirical loss  $\hat{\mathcal{L}}$  and relative error  $e$ , where  $i$  denotes the total iteration index along with the PF loops. In Figure 3, we have omitted the loss curve for DRM, since its value is negative. For all methods, the loss  $\hat{\mathcal{L}}$  and error  $e$  both decay steadily as the iteration proceeds, indicating stable convergence of the optimizer, but SEPINN enjoys the fastest decay and smallest error  $e$ , due to the improved regularity of  $w^*$ . The final error  $e$  saturates at around  $10^{-3}$  for SEPINN and  $10^{-2}$  for PINN, SAPINN, and FIPINN, but only  $10^{-1}$  for DRM. Indeed the accuracy of neural PDE solvers tends to stagnate at a level of  $10^{-2} \sim 10^{-3}$  [56, 65, 67, 17].

We next investigate a mixed boundary value problem [14, Example 1].

*Example 5.2.* The domain  $\Omega$  is the unit square  $\Omega = (0, 1)^2$ ,  $\Gamma_N = \{(x, 0) : x \in (0, \frac{1}{2})\}$ , and  $\Gamma_D = \partial\Omega \setminus \Gamma_N$ . The singular function  $s = r^{\frac{1}{2}} \sin \frac{\theta}{2}$  in the local polar coordinate  $(r, \theta)$  at  $(\frac{1}{2}, 0)$ . Set  $\rho = 1$  and  $R = \frac{1}{4}$  in (3.4), the source  $f = -\sin(\pi x)(-\pi^2 y^2(y-1) + 6y - 2) - \Delta(\eta_\rho s)$ . The exact solution  $u = \sin(\pi x)y^2(y-1) + \eta_\rho s$ , with the regular part  $w = \sin(\pi x)y^2(y-1)$  and stress intensity factor  $\gamma = 1$ .

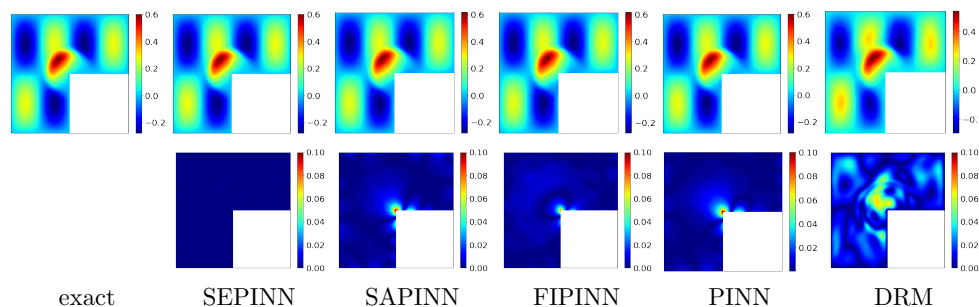


FIG. 2. The numerical approximations for Example 5.1 by SEPINN, SAPINN, FIPINN, PINN, and DRM. From top to bottom: DNN approximation and pointwise error.

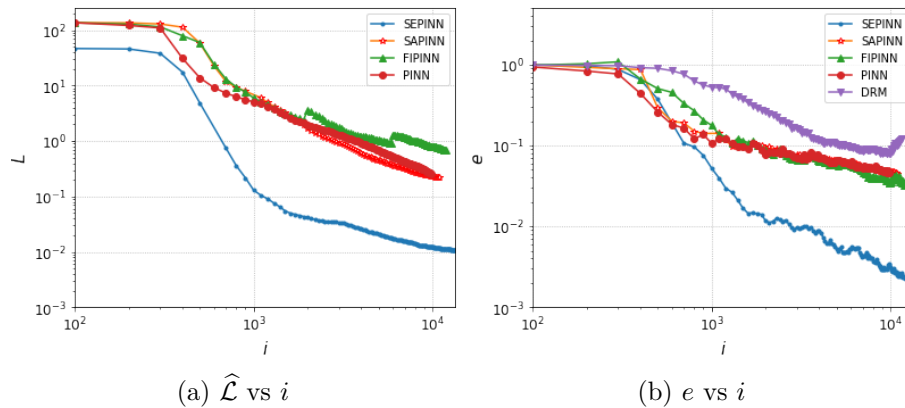


FIG. 3. Training dynamics for SEPINN and benchmark methods: (a) the decay of the empirical loss  $\widehat{\mathcal{L}}$  versus the iteration index  $i$  (counted along the PF trajectory) and (b) the error  $e$  versus  $i$ .

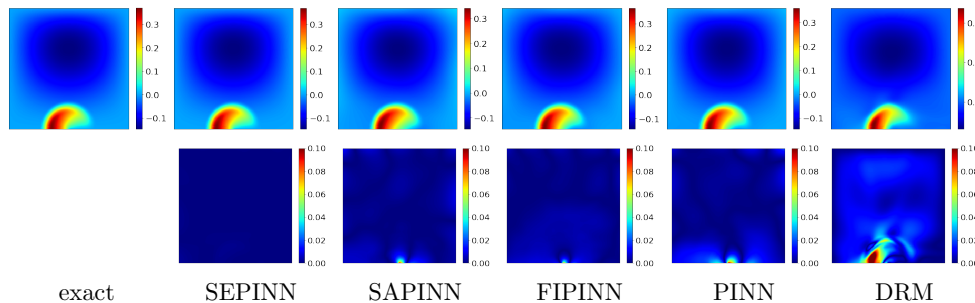


FIG. 4. The numerical approximations of Example 5.2 by the proposed SEPINN (error:  $4.62e-3$ ), SAPINN (error:  $4.40e-2$ ), FIPINN (error:  $3.65e-2$ ), PINN (error:  $7.33e-2$ ), and DRM (error:  $1.86e-1$ ). From top to bottom: DNN approximation and pointwise error.

This problem has a geometric singularity at the point  $(\frac{1}{2}, 0)$ , where the boundary condition changes from Dirichlet to Neumann with an interior angle  $\omega = \pi$ . We employ a 2-10-10-10-1 DNN (with 3 hidden layers, each having 10 neurons). The first stage of the PF strategy gives an estimate  $\widehat{\gamma}^* = 1.0033$ , and the prediction error  $e$  after the second stage is  $4.62e-3$ . The singularity at the crack point is accurately resolved by SEPINN; cf. Figure 4, which shows also the approximations by DRM and other PINN techniques. The overall solution accuracy is very similar to Example 5.1, and SEPINN achieves the smallest error. SAPINN and FIPINN improve the standard PINN, but still suffer from large errors near the singularity. Like before, the maximum error occurs near the boundary  $\partial\Omega$ , especially near the singularity point.

In addition, we have experimented with different architectures (depth, width) of the neural networks for the example. The training dynamics of the loss  $\widehat{\mathcal{L}}$  and the relative error  $e$  with four different neural networks are shown in Figure 5. The loss  $\widehat{\mathcal{L}}$  for the 2-20-20-20-20-20-1 network cannot stabilize at a lower level than the 2-20-20-20-1 network. Thus, increasing the size of the neural network or the number of sampling points alone does not necessarily lead to a smaller error for the approximation, as observed earlier in [36]. It remains to develop practical guidelines to select suitable architecture balancing good accuracy and computational expense.

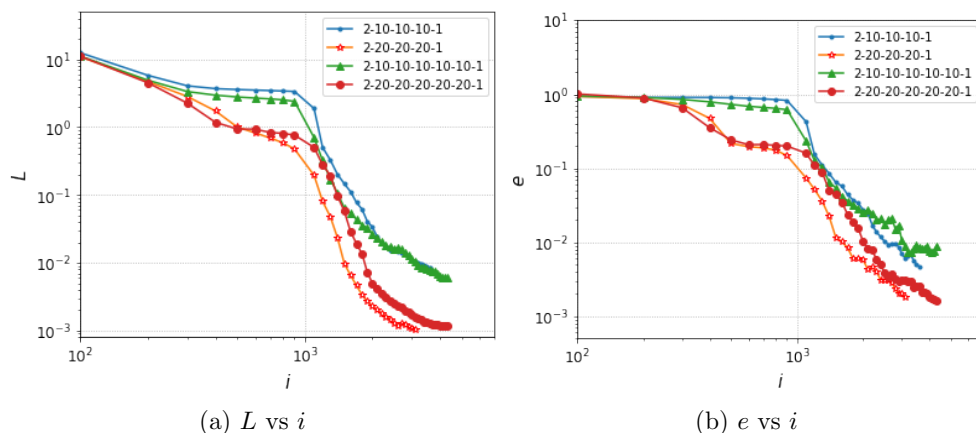


FIG. 5. The training dynamics for Example 5.2 with four different neural networks: (a) the loss  $L$  versus the iteration index  $i$  and (b) the error  $e$  versus the iteration index  $i$ .

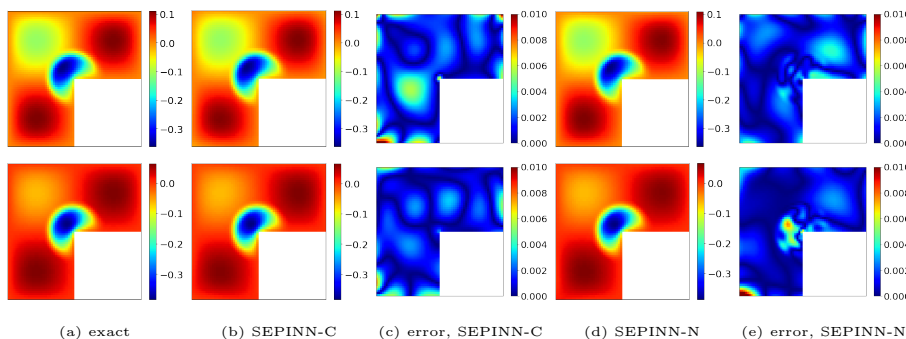


FIG. 6. The SEPINN-C and SEPINN-N approximations for Example 5.3, slices at  $z = \frac{1}{2}$  (top) and  $z = \frac{1}{4}$  (bottom).

The next example is a 3D Poisson equation adapted from [55, Example 1].

*Example 5.3.* Let  $\Omega_0 = (-1, 1)^2 \setminus ([0, 1] \times (-1, 0])$ , and the domain  $\Omega = \Omega_0 \times (-1, 1)$ . Define  $\Phi(r, z) = -2 \arctan \frac{e^{-\pi r} \sin \pi z}{1 + e^{-\pi r} \cos \pi z} = 2 \sum_{n=1}^{\infty} \frac{(-1)^n e^{-n\pi r} \sin n\pi z}{n}$ , and set  $\rho = 1$  and  $R = \frac{1}{2}$  in (3.4), the source  $f = 6x(y - y^3)(1 - z^2) + 6y(x - x^3)(1 - z^2) + 2(y - y^3)(x - x^3) - \Delta(\Phi \eta_\rho s)$ , with the singular function  $s = r^{\frac{2}{3}} \sin(\frac{2\theta}{3})$ , and a zero Dirichlet boundary condition. The exact solution  $u$  is given by  $u = (x - x^3)(y - y^3)(1 - z^2) + \Phi \eta_\rho s$ .

The coefficients  $\gamma_n^*$  are given by  $\gamma_0^* = 0$  and  $\gamma_n^* = (-1)^n \frac{2}{n}$  for  $n \in \mathbb{N}$ . In SEPINN-C (cutoff), we take a truncation level  $N = 20$  to approximate the first  $N + 1$  coefficients in the series for  $\Phi(r, z)$  and a 3-10-10-10-1 DNN to approximate  $w$ . In the first stage, we employ a learning rate  $1.0e-3$  for the DNN parameters  $\theta$  and  $8.0e-3$  for coefficients  $\gamma_N$ , and the prediction error  $e$  after the second stage is  $5.04e-2$ . We present the slices at  $z = \frac{1}{2}$  and  $z = \frac{1}{4}$  in Figure 6. The true and estimated values of  $\gamma_n$  are given in Table 5: the first few terms of the expansion are well approximated, but as the index  $n$  gets larger, the approximations of  $\hat{\gamma}_n^*$  become less accurate. However, the precise mechanism for the observation remains unclear. To offer insights, in Table 6, we show the relative errors  $e$  of  $\hat{w}$ ,  $e_S$  of the singular part  $\hat{S} = \hat{\gamma}_0 \eta_\rho s + \sum_{n=1}^N \hat{\gamma}_n e^{-\xi_n r} Z_n(z) \eta_\rho s$ , and  $e_u$  of the approximation  $\hat{u} = \hat{w} + \hat{S}$ , defined by  $e_u = \|u^* - \hat{u}\|_{L^2(\Omega)} / \|u^*\|_{L^2(\Omega)}$ , and

TABLE 5  
The estimated values of the parameters  $\gamma_n$  for Example 5.3, with five significant digits.

	$\gamma_0$	$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\gamma_6$
exact	0.000e0	-2.000e0	1.000e0	-6.667e-1	5.000e-1	-4.000e-1	3.333e-1
predicted	-7.793e-5	-2.000e0	1.006e0	-6.578e-1	5.135e-1	-4.238e-1	3.271e-1
	$\gamma_7$	$\gamma_8$	$\gamma_9$	$\gamma_{10}$	$\gamma_{11}$	$\gamma_{12}$	$\gamma_{13}$
exact	-2.857e-1	2.500e-1	-2.222e-1	2.000e-1	-1.818e-1	1.667e-1	-1.538e-1
predicted	-3.612e-1	2.410e-1	-2.384e-1	2.724e-1	-1.855e-1	3.700e-1	-1.045e-2
	$\gamma_{14}$	$\gamma_{15}$	$\gamma_{16}$	$\gamma_{17}$	$\gamma_{18}$	$\gamma_{19}$	$\gamma_{20}$
exact	1.429e-1	-1.333e-1	1.250e-1	-1.176e-1	1.111e-1	-1.053e-1	1.000e-1
predicted	2.761e-1	-2.082e-1	2.867e-1	1.239e-3	1.961e-1	7.774e-2	4.955e-1

TABLE 6  
Convergence of the SEPINN-C approximation with respect to the truncation level  $N$  for Example 5.3.

$N$	5	10	15	20
$e_{abs}$	1.76	0.797	1.42	0.937
$e$	7.14e-2	3.23e-2	5.77e-2	3.80e-2
$e_{u,abs}$	1.78	0.835	1.44	0.951
$e_u$	5.12e-2	2.40e-2	4.17e-2	2.73e-2
$e_{S,abs}$	0.625	0.271	0.245	0.146
$e_S$	2.55e-2	1.11e-2	1.00e-2	5.95e-3

likewise for  $e_S$ . The absolute errors are also given in order to give the full picture. Note that  $e$  and  $e_u$  exhibit very similar behavior, and both tend to stagnate, which is expected due to the slow decay of the coefficients  $\gamma_n^*$ . The error  $e_S$  decays steadily as  $N$  increases. Thus, the nonsteady convergence of  $e_u$  is due to the approximation of  $w$ , which might be due to the optimization error during the training.

Next we present the SEPINN-N approximation. We use two 4-layer DNNs, both of 3-10-10-10-1, for  $w$  and  $\Phi$ . The PF strategy is run with a maximum 2500 iterations for each fixed  $\sigma$ , and the final prediction error  $e$  is 3.80e-2. The maximum error of the SEPINN-N approximation is slightly smaller, and both can give excellent approximations.

Figure 7 compares the training dynamics for SEPINN-C and SEPINN-N. Figure 7(a) shows the convergence for the first few flux intensity factors  $\gamma$ , all initialized to 1, which are far from the optimal values. Nonetheless, the algorithm converges to the optimal one steadily. Within a few hundred iterations, the iterates approximate the exact one well. To accurately approximate  $w$ , more iterations are needed. Figure 7(b) shows the training dynamics for the DNN  $\Phi_\zeta$ , where  $e_\Phi = \|\Phi^* - \Phi_\zeta\|_{L^2(G)} / \|\Phi^*\|_{L^2(G)}$  ( $G$  is the support of the cutoff function  $\eta_\rho$  and  $\Phi^*$  is the exact one). We study the  $L^2(G)$  error instead of the  $L^2(\Omega)$  error, since  $\eta_\rho$  localizes its influence to  $G$ . The error  $e_\Phi$  eventually decreases to  $10^{-2}$ . The entire training process of the two methods is similar; cf. Figures 7(c) and (d). SEPINN-N takes more iterations than SEPINN-C, but SEPINN-C training takes longer: SEPINN-C requires evaluating the coefficient  $\gamma_n$ , which incurs taking the Laplacian of singular terms, whereas in SEPINN-N, all parameters are trained together.

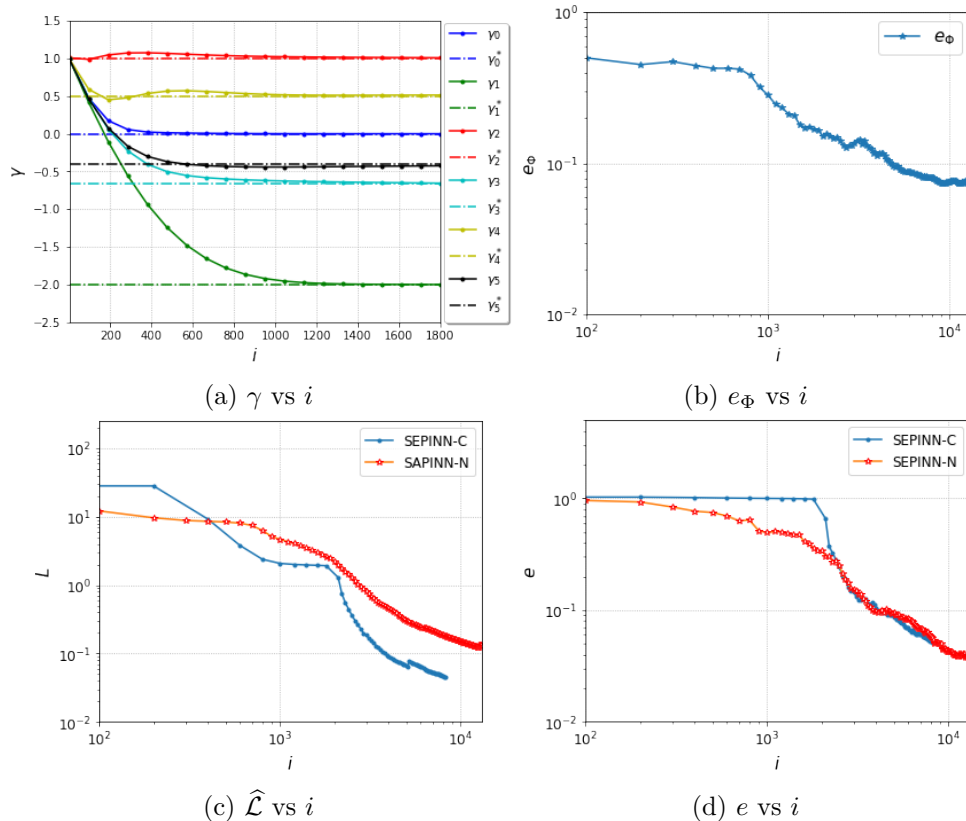


FIG. 7. The training dynamics of SEPINN-C and SEPINN-N: (a) the variation of first few coefficients versus iteration index  $i$ , (b) the error  $e_\Phi$  versus iteration index  $i$ , (c) the decay of the loss  $\widehat{\mathcal{L}}$  versus iteration index  $i$ , (d) the error  $e$  versus iteration index  $i$ .

The next example involves a combination of four singularities.

*Example 5.4.* Let the domain  $\Omega = (-\pi, \pi)^3$ ,  $\Gamma_D = \{(x, -\pi, z) : x \in (-\pi, 0), z \in (-\pi, \pi)\} \cup \{(x, \pi, z) : x \in (0, \pi), z \in (-\pi, \pi)\} \cup \{(0, y, z) : y \in (0, \pi), z \in (-\pi, \pi)\}$ , and  $\Gamma_N = \partial\Omega \setminus \Gamma_D$ . Let the vertices  $\mathbf{v}_1 : (0, -\pi)$ ,  $\mathbf{v}_2 : (\pi, 0)$ ,  $\mathbf{v}_3 : (0, \pi)$ , and  $\mathbf{v}_4 : (-\pi, 0)$ . This problem has four geometric singularities at boundary edges  $\mathbf{v}_j \times (-\pi, \pi)$ , where the type of the boundary condition changes from Dirichlet to Neumann with interior angles  $\omega_j = \pi$ ,  $j = 1, 2, 3, 4$ . Set  $\Phi_j(r_j, z) = r - \ln(2 \cosh r_j - 2 \cos z) = \sum_{n=1}^{\infty} \frac{2}{n} e^{-nr_j} \cos nz$ ,  $j = 1, 2, 3, 4$ , and set  $\rho_j = 1$  and  $R = \frac{1}{2}$  in (3.4), the source  $f = ((\frac{4}{\pi^2} - \frac{12x^2}{\pi^4})(1 - \frac{y^2}{\pi^2})^2 + (\frac{4}{\pi^2} - \frac{12y^2}{\pi^4})(1 - \frac{x^2}{\pi^2})^2 + (1 - \frac{x^2}{\pi^2})^2(1 - \frac{y^2}{\pi^2})^2) \cos z - \sum_{j=1}^4 \Delta(\Phi_j \eta_{\rho_j} s_j)$  with the singular functions  $s_j = r_j^{\frac{1}{2}} \cos(\frac{\theta_j}{2})$  for  $j = 1, 3$  and  $s_j = r_j^{\frac{1}{2}} \sin(\frac{\theta_j}{2})$  for  $j = 2, 4$ . The exact solution  $u$  is given by  $u = (1 - \frac{x^2}{\pi^2})^2(1 - \frac{y^2}{\pi^2})^2 \cos z + \sum_{j=1}^4 \Phi_j \eta_{\rho_j} s_j$ .

This example requires learning more parameters regardless of the method: for SEPINN-C, we have to expand four singular functions and learn their coefficients, whereas for SEPINN-N, we employ five networks to approximate  $w$  and  $\Phi_{\zeta, j}$  ( $j = 1, 2, 3, 4$ ). We take the number of sampling points  $N_d = 800$  and  $N_n = 1200$  on the boundaries  $\Gamma_D$  and  $\Gamma_N$ , respectively, and  $N_r = 10000$  in the domain  $\Omega$ .

First we present the SEPINN-C approximation. Note that  $\gamma_0^* = 0$  and  $\gamma_n^* = \frac{2}{n}$  for  $n \in \mathbb{N}$ . In SEPINN-C, we take a truncation level  $N = 15$  for all four singularities.

In the first stage, we take a learning rate  $2.0\text{e-}3$  for  $\theta$  and for coefficients  $r_1 = r_3 = 1.1\text{e-}2$ ,  $r_2 = r_4 = 7.0\text{e-}3$ . The final prediction error  $e$  is  $2.08\text{e-}2$ , and the estimated coefficients  $\hat{\gamma}^*$  are shown in Figure 8. The first few coefficients are well approximated, but the high-order ones are less accurate.

Next we show the SEPINN-N approximation, obtained with five 4-layer 3-10-10-10-1 DNNs to approximate  $w$  and  $\Phi_{\zeta,j}$  ( $j = 1, 2, 3, 4$ ) separately. The training process suffers from the following problem: using only L-BFGS tends to be trapped into a local minimum of the loss  $\hat{\mathcal{L}}_{\sigma}$ , which persists even after adjusting extensively the hyperparameters. Therefore, we first train the DNNs with Adam (learning rate  $r = 4.0\text{e-}3$ ) for 1000 iterations and then switch to L-BFGS (learning rate  $r = 0.2$ ) for a maximum 4000 iterations. This training strategy greatly improves the accuracy. The prediction error  $e$  after the second stage is  $3.83\text{e-}2$ . The approximation is fairly accurate but slightly less accurate than that by SEPINN-C in Figure 9 in both  $L^\infty(\Omega)$  and  $L^2(\Omega)$  norms.

Last, we illustrate SEPINN on the Laplacian eigenvalue problem.

*Example 5.5.* Let  $\Omega = (-1, 1)^2 \setminus ([0, 1] \times (-1, 0])$ . Consider the following Dirichlet Laplace eigenvalue problem:  $-\Delta u = \mu u$ , in  $\Omega$  and  $u = 0$  on  $\partial\Omega$ , where  $\mu > 0$  is the eigenvalue and  $u \neq 0$  is the corresponding eigenfunction.

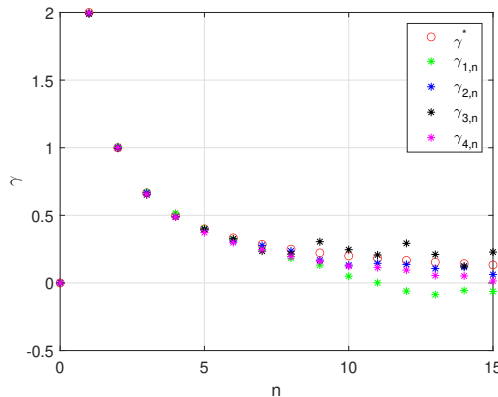


FIG. 8. The comparison between true and estimated values of the parameters  $\gamma_{j,n}$  for Example 5.4.

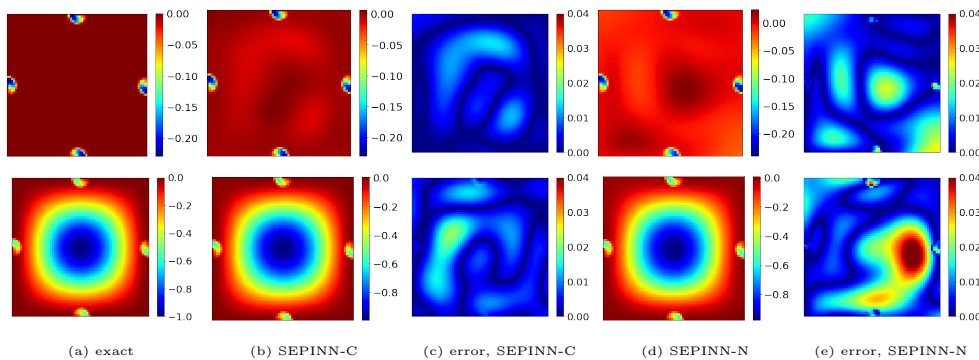


FIG. 9. The SEPINN-C and SEPINN-N approximations for Example 5.4, slices at  $z = \frac{\pi}{2}$  (top) and  $z = \pi$  (bottom).



TABLE 7  
The estimated eigenvalues by PINN and SEPINN for Example 5.5.

Eigenvalue	Reference [25]	SEPINN	Stress intensity factor	PINN
$\mu_1$	9.6397	9.6226	$\gamma_1 = 1.82\text{e-}2$	10.4840
$\mu_2$	15.1973	15.2100	$\gamma_2 = 8.71\text{e-}3$	15.0462

The eigenvalue problem on an L-shaped domain has been studied extensively [25, 46, 66]. The eigenfunctions may have singularity around the reentrant corner, but the analytic forms appear unavailable: the first eigenfunction  $u_1$  has a leading singular term  $r^{\frac{2}{3}} \sin(\frac{2}{3}\theta)$ , the second one  $u_2$  has  $r^{\frac{4}{3}} \sin(\frac{4}{3}\theta)$  [25], and the third  $u_3$  is analytic, given by  $u_3(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2)$ . We compute the first two leading eigenpairs  $(\mu_1, u_1)$  and  $(\mu_2, u_2)$ . The preceding discussions indicate  $u_1 \in H^1(\Omega)$  and  $u_2 \in H^2(\Omega)$ , and we split the leading singularity from both functions in order to benefit from SEPINN, i.e.,  $u_i = w_i + \gamma_i \eta_\rho s$ , with  $s = r^{\frac{2}{3}} \sin(\frac{2\theta}{3})$  and  $w_i$  is approximated by a DNN. Following the ideas in [8] and SEPINN, we employ the loss

$$\begin{aligned} \mathcal{L}_\sigma(w_1, w_2; \gamma_1, \gamma_2) = & \sum_{i=1}^2 \left( \|\Delta(w_i + \gamma_i \eta_\rho s) + \psi(u_i)(w_i + \gamma_i \eta_\rho s)\|_{L^2(\Omega)}^2 + \sigma_1 \|w_i\|_{L^2(\partial\Omega)}^2 \right. \\ & + \alpha \left| \|w_i + \gamma_i \eta_\rho s\|_{L^2(\Omega)}^2 - 1 \right| + \nu_i \psi(u_i) \\ & \left. + \beta \left| (w_1 + \gamma_1 \eta_\rho s, w_2 + \gamma_2 \eta_\rho s)_{L^2(\Omega)} \right|, \right. \end{aligned}$$

where  $\alpha$ ,  $\nu_i (i = 1, 2)$  and  $\beta$  are hyperparameters and  $\psi(u_i)$  is the Rayleigh quotient,

$$(5.2) \quad \psi(u_i) = \|\nabla u_i\|_{L^2(\Omega)}^2 / \|u_i\|_{L^2(\Omega)}^2, \quad i = 1, 2,$$

which estimates the eigenvalue  $\mu_i$  using the eigenfunction  $u_i$ , by Rayleigh's principle. We employ an alternating iteration method: we first approximate the eigenfunction  $u_i$  by minimizing the loss  $\mathcal{L}_\sigma$  and then update the eigenvalue  $\mu_i$  by (5.2), which is then substituted back into  $\mathcal{L}_\sigma$ . These two steps are repeated until convergence.

In SEPINN, we employ two 2-10-10-10-10-10-1 DNNs to approximate the regular parts  $w_1$  and  $w_2$ , and take  $\alpha = 100$ ,  $\beta = 135$ ,  $\nu_1 = 0.02$ , and  $\nu_2 = 0.01$  and determine the parameter  $\sigma_d$  by the PF strategy. We use Adam with a learning rate  $2\text{e-}3$  for all DNN parameters and  $\gamma$ . Table 7 shows that singularity enrichment helps solve the eigenvalue problem. Indeed, we can approximate  $u_1$  better and get more accurate eigenvalue estimates. Note that during the training process of the standard PINN, the DNN approximation actually directly approaches  $u_2$  and cannot capture  $u_1$ , since it cannot be resolved accurately. Even with a larger penalty  $\nu_1$  for the first eigenvalue, this does not improve the accuracy, while  $u_2$  can be approximated well due to the better regularity of  $u_2$ , i.e.,  $u_2 \in H^2(\Omega)$ .

**6. Conclusions.** In this work, we have developed a family of novel neural solvers for second-order elliptic boundary value problems with geometric singularities, e.g., corner singularity and mixed boundary conditions in the 2D case, and edge singularities in the 3D case. The basic idea is to enrich the ansatz space spanned by deep neural networks, by incorporating suitable singular functions. These singular functions are designed to capture leading singularities of the exact solution. In so doing, we can obtain more accurate approximations. We discuss several variants of the method, depending on the specific scenarios, and discuss the extensions to the eigenvalue problem. Additionally, we provide preliminary theoretical guarantees of

the approach. Numerical experiments indicate that the approach is indeed highly effective and flexible and works for a broad range of problem settings.

There are several directions for further research. First, one interesting question is to extend the approach to more general singularities, e.g., conic singularities, singularities along curved edges, rotating solids  $\Omega = \Omega_0 \times (0, l)$  (with  $\Omega_0$  depending on the height), and higher-dimensional cases, as well as more complex problems, e.g., the Maxwell system and the linear elasticity system. The key is to identify the singularity functions. Note that in principle, the singularity functions can be approximate, e.g., computed using FEM. Second, it is of importance to study more closely the optimization aspects of the approach. It is already challenging for standard neural solvers [43, 63], and the issue appears even more delicate for SEPINN, as indicated by the numerical experiments. Third, it is of much interest to design techniques that allow imposing the boundary conditions exactly, which would facilitate deriving stronger and better estimates.

**Appendix A. Technical preliminaries.** In this appendix, we collect several preliminary results that are used in the error analysis of the proposed SEPINN. The Rademacher complexity [1, 6] measures the complexity of a collection of functions by the correlation between function values with Rademacher random variables, i.e., with probability  $P(\omega = 1) = P(\omega = -1) = \frac{1}{2}$ .

**DEFINITION A.1.** Let  $\mathcal{F}$  be a real-valued function class defined on the domain  $D$  and let  $\xi = \{\xi_j\}_{j=1}^n$  be i.i.d. samples from the distribution  $\mathcal{U}(D)$ . Then the Rademacher complexity  $\mathfrak{R}_n(\mathcal{F})$  of  $\mathcal{F}$  is defined by

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\xi, \omega} \left[ \sup_{f \in \mathcal{F}} n^{-1} \left| \sum_{j=1}^n \omega_j f(\xi_j) \right| \right],$$

where  $\omega = \{\omega_j\}_{j=1}^n$  are i.i.d. Rademacher random variables.

Then we have the following PAC-type generalization bound [52, Theorem 3.1].

**LEMMA A.2.** Let  $X_1, \dots, X_n$  be a set of i.i.d. random variables, and let  $\mathcal{F}$  be a function class defined on  $D$  such that  $\sup_{f \in \mathcal{F}} \|f\|_{L^\infty(D)} \leq M_{\mathcal{F}} < \infty$ . Then for any  $\tau \in (0, 1)$ , with probability at least  $1 - \tau$ ,

$$\sup_{f \in \mathcal{F}} \left| n^{-1} \sum_{j=1}^n f(X_j) - \mathbb{E}[f(X)] \right| \leq 2\mathfrak{R}_n(\mathcal{F}) + 2M_{\mathcal{F}} \sqrt{\frac{\log \frac{1}{\tau}}{2n}}.$$

To apply Lemma A.2, we bound Rademacher complexities of the function classes  $\mathcal{H}_r$ ,  $\mathcal{H}_d$ , and  $\mathcal{H}_n$ . This follows from Dudley's formula in Lemma A.6. The next lemma gives useful boundedness and Lipschitz continuity of the DNN function class in terms of  $\theta$ ; see [36, Lemma 3.4 and Remark 3.3] and [37, Lemma 5.3]. The estimates also hold when  $L^\infty(\Omega)$  is replaced with  $L^\infty(\Gamma_D)$  or  $L^\infty(\Gamma_N)$ . The notation  $L^\infty(\Omega; \mathbb{R}^d)$  denotes the  $L^\infty(\Omega)$  norm for  $\mathbb{R}^d$ -valued functions. Note that  $\|v_\theta\|_{L^\infty(\Omega)}$  is independent of the depth  $L$  since the activation function  $\varrho$  satisfies  $\|\varrho\|_{L^\infty(\mathbb{R})} \leq 1$ .

**LEMMA A.3.** Let  $L$ ,  $W$ , and  $B_\theta$  be the depth, width, and maximum weight bound of a DNN function class  $\mathcal{W}$  with  $N_\theta$  nonzero weights. Then for any  $v_\theta \in \mathcal{W}$ , the following estimates hold:

- (i)  $\|v_\theta\|_{L^\infty(\Omega)} \leq WB_\theta$ ,  $\|v_\theta - v_{\tilde{\theta}}\|_{L^\infty(\Omega)} \leq 2LW^L B_\theta^{L-1} |\theta - \tilde{\theta}|_{\ell^\infty}$ ;
- (ii)  $\|\nabla v_\theta\|_{L^\infty(\Omega; \mathbb{R}^d)} \leq \sqrt{2}W^{L-1} B_\theta^L$ ,  $\|\nabla(v_\theta - v_{\tilde{\theta}})\|_{L^\infty(\Omega; \mathbb{R}^d)} \leq \sqrt{2}L^2 W^{2L-2} B_\theta^{2L-2} |\theta - \tilde{\theta}|_{\ell^\infty}$ ;

$$(iii) \|\Delta v_\theta\|_{L^\infty(\Omega)} \leq 2LW^{2L-2}B_\theta^{2L}, \|\Delta(v_\theta - v_{\tilde{\theta}})\|_{L^\infty(\Omega)} \leq 8N_\theta L^2 W^{3L-3} B_\theta^{3L-3} |\theta - \tilde{\theta}|_{\ell^\infty}.$$

Lemma A.3 implies the uniform boundedness and Lipschitz continuity of functions in  $\mathcal{H}_r$ ,  $\mathcal{H}_d$ , and  $\mathcal{H}_n$ .

LEMMA A.4. *There exists a constant  $c$  depending on  $\|f\|_{L^\infty(\Omega)}$ ,  $\|\Delta(\eta_\rho s)\|_{L^\infty(\Omega)}$ , and  $B_\gamma$  such that*

$$\begin{aligned} \|h(w_\theta, \gamma)\|_{L^\infty(\Omega)} &\leq cL^2 W^{4L-4} B_\theta^{4L} \quad \forall h \in \mathcal{H}_r, \\ \|h(w_\theta)\|_{L^\infty(\Gamma_D)} &\leq cW^2 B_\theta^2 \quad \forall h \in \mathcal{H}_d, \\ \|h(w_\theta)\|_{L^\infty(\Gamma_N)} &\leq cW^{2L-2} B_\theta^{2L} \quad \forall h \in \mathcal{H}_n. \end{aligned}$$

Moreover, the following Lipschitz continuity estimates hold:

$$\begin{aligned} \|h(w_\theta, \gamma) - \tilde{h}(w_{\tilde{\theta}}, \tilde{\gamma})\|_{L^\infty(\Omega)} &\leq cN_\theta L^3 W^{5L-5} B_\theta^{5L-3} (|\theta - \tilde{\theta}|_{\ell^\infty} + |\gamma - \tilde{\gamma}|) \quad \forall h, \tilde{h} \in \mathcal{H}_r, \\ \|h(w_\theta) - h(w_{\tilde{\theta}})\|_{L^\infty(\Gamma_D)} &\leq cLW^{L+1} B_\theta^L |\theta - \tilde{\theta}|_{\ell^\infty} \quad \forall h, \tilde{h} \in \mathcal{H}_d, \\ \|h(w_\theta) - \tilde{h}(w_{\tilde{\theta}})\|_{L^\infty(\Gamma_N)} &\leq cL^2 W^{3L-3} B_\theta^{3L-2} |\theta - \tilde{\theta}|_{\ell^\infty} \quad \forall h, \tilde{h} \in \mathcal{H}_n. \end{aligned}$$

*Proof.* All the estimates are direct from Lemma A.3.  $\square$

Next, we state Dudley's lemma [49, Theorem 9], [64, Theorem 1.19], which bounds Rademacher complexities using covering number.

DEFINITION A.5. *Let  $(\mathcal{M}, m)$  be a metric space of real valued functions, and  $\mathcal{G} \subset \mathcal{M}$ . A set  $\{x_i\}_{i=1}^n \subset \mathcal{G}$  is called an  $\epsilon$ -cover of  $\mathcal{G}$  if for any  $x \in \mathcal{G}$ , there exists an element  $x_i \in \{x_i\}_{i=1}^n$  such that  $m(x, x_i) \leq \epsilon$ . The  $\epsilon$ -covering number  $\mathcal{C}(\mathcal{G}, m, \epsilon)$  is the minimum cardinality among all  $\epsilon$ -covers of  $\mathcal{G}$  with respect to  $m$ .*

LEMMA A.6. *Let  $M_{\mathcal{F}} := \sup_{f \in \mathcal{F}} \|f\|_{L^\infty(\Omega)}$ , and let  $\mathcal{C}(\mathcal{F}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$  be the covering number of  $\mathcal{F}$ . Then the Rademacher complexity  $\mathfrak{R}_n(\mathcal{F})$  is bounded by*

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

**Reproducibility of computational results.** This paper has been awarded the ‘‘SIAM Reproducibility Badge: Code and data available’’ as a recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at <https://github.com/hhjc-web/SEPINN.git> and in the supplementary materials (M160119\_SuppMat.pdf [local/web 399KB]).

**Acknowledgment.** The authors are grateful to two anonymous referees for their many constructive comments, which have led to a significant improvement of the quality of the paper.

#### REFERENCES

- [1] M. ANTHONY AND P. L. BARTLETT, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 1999, <https://doi.org/10.1017/CBO9780511624216>.
- [2] T. APEL AND S. NICAISE, *The finite element method with anisotropic mesh grading for elliptic problems in domains with corners and edges*, Math. Methods Appl. Sci., 21 (1998), pp. 519–549, [https://doi.org/10.1002/\(SICI\)1099-1476\(199804\)21:6<519::AID-MMA962>3.3.CO;2-I](https://doi.org/10.1002/(SICI)1099-1476(199804)21:6<519::AID-MMA962>3.3.CO;2-I).

- [3] T. APEL, A.-M. SÄNDIG, AND J. R. WHITEMAN, *Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains*, Math. Methods Appl. Sci., 19 (1996), pp. 63–85, [https://doi.org/10.1002/\(SICI\)1099-1476\(19960110\)19:1<63::AID-MMA764>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1099-1476(19960110)19:1<63::AID-MMA764>3.0.CO;2-S).
- [4] F. ASSOUS, P. CIARLET, JR., AND J. SEGRÉ, *Numerical solution to the time-dependent Maxwell equations in two-dimensional singular domains: The singular complement method*, J. Comput. Phys., 161 (2000), pp. 218–249, <https://doi.org/10.1006/jcph.2000.6499>.
- [5] I. BABUŠKA AND A. MILLER, *The post-processing approach in the finite element method – part 2: The calculation of stress intensity factors*, Internat. J. Numer. Methods Engrg., 20 (1984), pp. 1111–1129.
- [6] P. L. BARTLETT AND S. MENDELSON, *Rademacher and Gaussian complexities: Risk bounds and structural results*, J. Mach. Learn. Res., 3 (2002), pp. 463–482, <https://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf>.
- [7] A. G. BAYDIN, B. A. PEARLMUTTER, A. A. RADUL, AND J. M. SISKIND, *Automatic differentiation in machine learning: A survey*, J. Mach. Learn. Res., 18 (2018), pp. 1–43.
- [8] I. BEN-SHAUL, L. BAR, D. FISHELOV, AND N. SOCHEN, *Deep learning solution of the eigenvalue problem for differential operators*, Neural Comput., 35 (2023), pp. 1100–1134, <https://doi.org/10.1162/neco.a.01583>.
- [9] M. BERGGREN, *Approximations of very weak solutions to boundary-value problems*, SIAM J. Numer. Anal., 42 (2004), pp. 860–877, <https://doi.org/10.1137/S0036142903382048>.
- [10] H. BLUM AND M. DOBROWOLSKI, *On finite element methods for elliptic equations on domains with corners*, Computing, 28 (1982), pp. 53–63, <https://doi.org/10.1007/BF02237995>.
- [11] R. H. BYRD, P. LU, J. NOCEDAL, AND C. Y. ZHU, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208, <https://doi.org/10.1137/0916069>.
- [12] Z. CAI, J. CHEN, AND M. LIU, *Least-squares ReLU neural network (LSNN) method for linear advection-reaction equation*, J. Comput. Phys., 443 (2021), 110514, <https://doi.org/10.1016/j.jcp.2021.110514>.
- [13] Z. CAI AND S. KIM, *A finite element method using singular functions for the Poisson equation: Corner singularities*, SIAM J. Numer. Anal., 39 (2001), pp. 286–299.
- [14] Z. CAI, S. KIM, S. KIM, AND S. KONG, *A finite element method using singular functions for poisson equations: Mixed boundary conditions*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 2635–2648.
- [15] Z. CAI, S. KIM, AND B.-C. SHIN, *Solution methods for the Poisson equation with corner singularities: Numerical results*, SIAM J. Sci. Comput., 23 (2001), pp. 672–682.
- [16] F. CUCKER AND S. SMALE, *On the mathematical foundations of learning*, Bull. Amer. Math. Soc. (N.S.), 39 (2002), pp. 1–49, <https://doi.org/10.1090/S0273-0979-01-00923-5>.
- [17] S. CUOMO, V. SCHIANO DI COLA, F. GIAMPAOLO, G. ROZZA, M. RAISSI, AND F. PICCIALLI, *Scientific machine learning through physics-informed neural networks: Where we are and what’s next*, J. Sci. Comput., 92 (2022), 88, <https://doi.org/10.1007/s10915-022-01939-z>.
- [18] M. DAUGE, *Elliptic Boundary Value Problems on Corner Domains*, Springer-Verlag, Berlin, 1988.
- [19] M. DAUGE, R. BOYER, E. CROC, Y. DERMENJIAN, F. HUBERT, AND E. PRATT, *Singularities of corner problems and problems of corner singularities*, ESAIM Proc., 6 (1999), pp. 19–40.
- [20] T. DE RYCK, A. D. JAGTAP, AND S. MISHRA, *Error estimates for physics-informed neural networks approximating the Navier–Stokes equations*, IMA J. Numer. Anal., 44 (2024), pp. 83–119, <https://doi.org/10.1093/imanum/drac085>.
- [21] T. DE RYCK AND S. MISHRA, *Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs*, Adv. Comput. Math., 48 (2022), 79, <https://doi.org/10.1007/s10444-022-09985-9>.
- [22] P. DESTUYNDER AND M. DJAOUA, *Estimation de l’erreur sur le coefficient de la singularité de la solution d’un problème elliptique sur un ouvert avec coin*, RAIRO Anal. Numér., 14 (1980), pp. 239–248.
- [23] W. E, J. HAN, AND A. JENTZEN, *Algorithms for solving high dimensional PDEs: From non-linear Monte Carlo to machine learning*, Nonlinearity, 35 (2022), pp. 278–310, <https://doi.org/10.1088/1361-6544/ac337f>.
- [24] G. J. FIX, S. GULATI, AND G. I. WAKOFF, *On the use of singular functions with finite element approximations*, J. Comput. Phys., 13 (1973), pp. 209–228, [https://doi.org/10.1016/0021-9991\(73\)90023-5](https://doi.org/10.1016/0021-9991(73)90023-5).
- [25] L. FOX, P. HENRICI, AND C. B. MOLER, *Approximations and bounds for eigenvalues of elliptic operators*, SIAM J. Numer. Anal., 4 (1967), pp. 89–102, <https://doi.org/10.1137/0704008>.

- [26] T.-P. FRIES AND T. BELYTSCHKO, *The extended/generalized finite element method: An overview of the method and its applications*, Internat. J. Numer. Methods Engrg., 84 (2010), pp. 253–304, <https://doi.org/10.1002/nme.2914>.
- [27] Z. GAO, T. TANG, L. YAN, AND T. ZHOU, *Failure-informed adaptive sampling for pinns, part II: Combining with re-sampling and subset simulation*, Commun. Appl. Math. Comput., (2023), <https://doi.org/10.1007/s42967-023-00312-7>.
- [28] Z. GAO, L. YAN, AND T. ZHOU, *Failure-informed adaptive sampling for PINNs*, SIAM J. Sci. Comput., 45 (2023), pp. A1971–A1994.
- [29] F. D. GASPOZ AND P. MORIN, *Convergence rates for adaptive finite elements*, IMA J. Numer. Anal., 29 (2009), pp. 917–936, <https://doi.org/10.1093/imanum/drn039>.
- [30] P. GRISVARD, *Elliptic Problems in Nonsmooth Domains*, SIAM, Philadelphia, 2011.
- [31] I. GUHRING AND M. RASLAN, *Approximation rates for neural networks with encodable weights in smoothness spaces*, Neural Networks, 134 (2021), pp. 107–130.
- [32] T. HU, B. JIN, AND Z. ZHOU, *Solving elliptic problems with singular sources using singularity splitting deep Ritz method*, SIAM J. Sci. Comput., 45 (2023), pp. A2043–A2074.
- [33] X. HUANG, H. LIU, B. SHI, Z. WANG, K. YANG, Y. LI, B. WENG, M. WANG, H. CHU, J. ZHOU, F. YU, B. HUA, B. DONG, AND L. CHEN, *Solving partial differential equations with point source based on physics-informed neural networks*, in Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI-22), 2022, pp. 3839–3846.
- [34] Y. JIAO, Y. LAI, D. LI, X. LU, F. WANG, Y. WANG, AND J. Z. YANG, *A rate of convergence of physics informed neural networks for the linear second order elliptic PDEs*, Commun. Comput. Phys., 31 (2022), pp. 1272–1295, <https://doi.org/10.4208/cicp.oa-2021-0186>.
- [35] Y. JIAO, Y. LAI, Y. LO, Y. WANG, AND Y. YANG, *Error analysis of deep ritz methods for elliptic equations*, Anal. Appl. (Singap.), 22 (2024), pp. 57–87, <https://doi.org/10.1142/S021953052350015X>.
- [36] B. JIN, X. LI, AND X. LU, *Imaging conductivity from current density magnitude using neural networks*, Inverse Problems, 38 (2022), 075003, <https://doi.org/10.1088/1361-6420/ac6d03>.
- [37] B. JIN, R. SAU, L. YIN, AND Z. ZHOU, *Solving Elliptic Optimal Control Problems Using Physics Informed Neural Networks*, preprint, arXiv:2308.11925, 2023.
- [38] G. E. KARNIADAKIS, I. G. KEVREKIDIS, L. LU, P. PERDIKARIS, S. WANG, AND L. YANG, *Physics-informed machine learning*, Nature Rev. Phys., 3 (2021), pp. 422–440.
- [39] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, in Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015.
- [40] V. A. KOZLOV, *Singularities of solutions of the Dirichlet problem for elliptic equations in a neighborhood of corner points*, Algebra i Analiz, 1 (1989), pp. 161–177.
- [41] V. A. KOZLOV, V. G. MAZ'YA, AND J. ROSSMANN, *Elliptic Boundary Value Problems in Domains with Point Singularities*, AMS, Providence, RI, 1997, <https://doi.org/10.1090/surv/052>.
- [42] V. A. KOZLOV, V. G. MAZ'YA, AND J. ROSSMANN, *Spectral Problems Associated with Corner Singularities of Solutions to Elliptic Equations*, AMS, Providence, RI, 2001, <https://doi.org/10.1090/surv/085>.
- [43] A. KRISHNAPRIYAN, A. GHOLAMI, S. ZHE, R. KIRBY, AND M. W. MAHONEY, *Characterizing possible failure modes in physics-informed neural networks*, in Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 26548–26560.
- [44] I. E. LAGARIS, A. LIKAS, AND D. I. FOTIADIS, *Artificial neural networks for solving ordinary and partial differential equations*, IEEE Trans. Neural Networks, 9 (1998), pp. 987–1000.
- [45] M.-N. LE ROUX, *Équations intégrales pour le problème du potentiel électrique dans le plan*, Samuel Beckett Today/aujourd'hui, 61 (2008), pp. 133–141.
- [46] X. LIU AND S. OISHI, *Verified eigenvalue evaluation for the Laplacian over polygonal domains of arbitrary shape*, SIAM J. Numer. Anal., 51 (2013), pp. 1634–1654, <https://doi.org/10.1137/120878446>.
- [47] L. LU, R. PESTOURIE, W. YAO, Z. WANG, F. VERDUGO, AND S. G. JOHNSON, *Physics-informed neural networks with hard constraints for inverse design*, SIAM J. Sci. Comput., 43 (2021), pp. B1105–B1132, <https://doi.org/10.1137/21M1397908>.
- [48] Y. LU, H. CHEN, J. LU, L. YING, AND J. BLANCHET, *Machine learning for elliptic PDEs: Fast rate generalization bound, neural scaling law and minimax optimality*, in Proceedings of the International Conference on Learning Representations, 2022, <https://openreview.net/forum?id=mhYUBYN0Gz>.
- [49] Y. LU, J. LU, AND M. WANG, *A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations*, in Proceedings of the Conference on Learning Theory, PMLR, 2021, pp. 3196–3241.

- [50] V. MAZ'YA AND J. ROSSMANN, *Elliptic Equations in Polyhedral Domains*, AMS, Providence, RI, 2010, <https://doi.org/10.1090/surv/162>.
- [51] S. MISHRA AND R. MOLINARO, *Estimates on the generalization error of physics-informed neural networks for approximating PDEs*, IMA J. Numer. Anal., 43 (2023), pp. 1–43, <https://doi.org/10.1093/imanum/drab093>.
- [52] M. MOHRI, A. ROSTAMIZADEH, AND A. TALWALKAR, *Foundations of Machine Learning*, MIT Press, Cambridge, MA, 2018.
- [53] J. MÜLLER AND M. ZEINHOFER, *Error estimates for the deep Ritz method with boundary penalty*, in Proceedings of the 3rd Annual Conference on Mathematical and Scientific Machine Learning, 2022.
- [54] B. NKEMZI AND M. JUNG, *Flux intensity functions for the Laplacian at polyhedral edges*, Int. J. Fracture, 175 (2012), pp. 167–185.
- [55] B. NKEMZI AND M. JUNG, *Singular solutions of the Poisson equation at edges of three-dimensional domains and their treatment with a predictor–corrector finite element method*, Numer. Methods Partial Differential Equations, 37 (2021), pp. 836–853.
- [56] M. RAISSI, P. PERDIKARIS, AND G. E. KARNIADAKIS, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707.
- [57] G. RAUGEL, *Résolution numérique par une méthode d'éléments finis du problème de Dirichlet pour le laplacien dans un polygone*, C. R. Acad. Sci. Paris Sér. A-B, 286 (1978), pp. A791–A794.
- [58] Y. SHIN, J. DARBON, AND G. E. KARNIADAKIS, *On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs*, Commun. Comput. Phys., 28 (2020), pp. 2042–2074, <https://doi.org/10.4208/cicp.0a-2020-0193>.
- [59] J. W. SIEGEL, Q. HONG, X. JIN, W. HAO, AND J. XU, *Greedy training algorithms for neural networks and applications to PDEs*, J. Comput. Phys., 484 (2023), 112084, <https://doi.org/10.1016/j.jcp.2023.112084>.
- [60] J. SIRIGNANO AND K. SPILIOPOULOS, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys., 375 (2018), pp. 1339–1364.
- [61] E. P. STEPHAN AND J. R. WHITEMAN, *Singularities of the Laplacian at corners and edges of three-dimensional domains and their treatment with finite element methods*, Math. Methods Appl. Sci., 10 (1988), pp. 339–350, <https://doi.org/10.1002/mma.1670100309>.
- [62] N. SUKUMAR AND A. SRIVASTAVA, *Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks*, Comput. Methods Appl. Mech. Engrg., 389 (2022), 114333, <https://doi.org/10.1016/j.cma.2021.114333>.
- [63] S. WANG, X. YU, AND P. PERDIKARIS, *When and why PINNs fail to train: A neural tangent kernel perspective*, J. Comput. Phys., 449 (2022), 110768, <https://doi.org/10.1016/j.jcp.2021.110768>.
- [64] M. M. WOLF, *Mathematical Foundations of Supervised Learning*, Lecture notes, [https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MA4801\\_2021S/ML.pdf](https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MA4801_2021S/ML.pdf), 2021.
- [65] B. YU AND W. E, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat., 6 (2018), pp. 1–12.
- [66] Q. YUAN AND Z. HE, *Bounds to eigenvalues of the Laplacian on L-shaped domain by variational methods*, J. Comput. Appl. Math., 233 (2009), pp. 1083–1090, <https://doi.org/10.1016/j.cam.2009.08.114>.
- [67] Y. ZANG, G. BAO, X. YE, AND H. ZHOU, *Weak adversarial networks for high-dimensional partial differential equations*, J. Comput. Phys., 411 (2020), 109409, <https://doi.org/10.1016/j.jcp.2020.109409>.