

Error analysis for finite element operator learning methods for solving parametric second-order elliptic PDEs

Youngjoon Hong*, Seungchan Ko†, and Jaeyong Lee‡

Abstract

In this paper, we provide a theoretical analysis of a type of operator learning method without data reliance based on the classical finite element approximation, which is called the finite element operator network (FEONet). We first establish the convergence of this method for general second-order linear elliptic PDEs with respect to the parameters for neural network approximation. In this regard, we address the role of the condition number of the finite element matrix in the convergence of the method. Secondly, we derive an explicit error estimate for the self-adjoint case. For this, we investigate some regularity properties of the solution in certain function classes for a neural network approximation, verifying the sufficient condition for the solution to have the desired regularity. Finally, we will also conduct some numerical experiments that support the theoretical findings, confirming the role of the condition number of the finite element matrix in the overall convergence.

Keywords: unsupervised operator learning, convergence analysis, complex geometry, finite element method, condition number, approximation error, generalization error, Rademacher complexity

AMS Classification: 65N30, 65M60, 65N12, 68T07, 68U07

1 Introduction

The emerging field of scientific machine learning, which bridges the gap between traditional numerical analysis and machine learning, has introduced innovative approaches that enrich conventional numerical methods, especially in tackling complex tasks. At the forefront of this evolution is the field of physics-informed neural networks (PINNs) [45]. PINNs employ neural networks trained to comprehend the underlying physics of systems, thereby enhancing the capability to solve PDEs for physics-based problems using neural networks. This advancement has spurred the development of various PINN variants [31, 52]. However, these variants are limited by their need for retraining with each new set of input data, such as initial conditions and boundary conditions, which hampers their utility in dynamic systems where real-time predictions are essential.

Addressing this limitation, operator networks have emerged, employing data-driven approaches to understand mathematical operators in physical systems, particularly for parametric PDEs [33, 6]. A significant breakthrough in this area is the Deep Operator Network (DeepONet) architecture [36], founded on the universal approximation theorem for operators. DeepONets facilitate rapid solution prediction when PDE data varies, but depends on extensive pre-computed training data pairs, which is a demanding task, especially for complex or nonlinear systems. To mitigate these issues, hybrid models like Physics-Informed Neural Operator (PINO) [34] and Physics-Informed DeepONet (PIDeepONet) [50] have been introduced. These models amalgamate the strengths of PINNs and operator learning by embedding physical equations within the loss function of neural operators. Despite this innovation, they often face challenges such as reduced accuracy in complex geometries, difficulty managing stiff problems, and

*Department of Mathematical Sciences, KAIST, Daejeon, Republic of Korea. Email: hongyj@kaist.ac.kr

†Department of Mathematics, Inha University, Incheon, Republic of Korea. Email: scko@inha.ac.kr

‡Department of AI, Chung-Ang University, Seoul, Republic of Korea. Email: jaeyong@cau.ac.kr

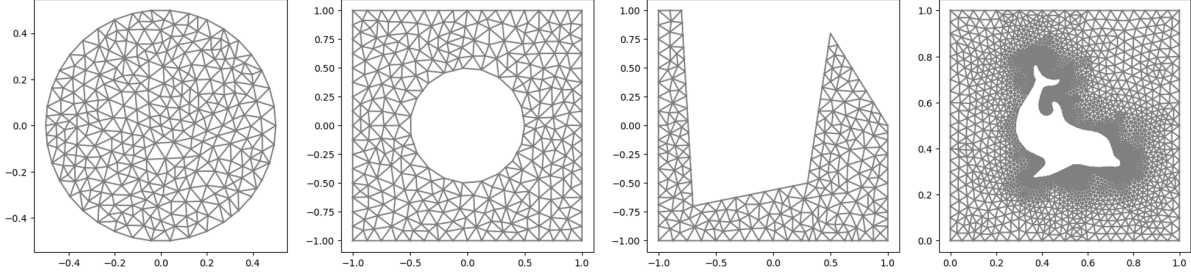


Figure 1: Examples of complex domains where FEONet can predict solutions.

significant generalization errors due to limited input data [24, 11, 30]. Furthermore, employing neural networks as the solution space complicates the imposition of various boundary values, consequently impacting the precision of solutions [9].

In response to these limitations, a novel unsupervised operator network based on finite element methods, termed the Finite Element Operator Network (FEONet), was developed [32]. The main focus of this paper is on the extensive error analysis of FEONet. In the finite element method (FEM) framework, the numerical solution $u_h(x)$ is approximated as a linear combination of nodal coefficients α_k , and nodal basis functions, $\phi_k(x)$, defined by piecewise polynomials over a mesh. This is represented as $u_h(x) = \sum \alpha_k \phi_k(x)$, $x \in \mathbb{R}^d$. Building on this concept, the FEONet predicts PDE solutions under various inputs like initial conditions and boundary conditions. It is versatile and able to handle multiple PDE instances across complex domains as shown in Figure 1 without data reliance. The loss function of FEONet, inspired by the classical FEM, is based on the residual of the finite element approximation, ensuring accurate PDE solutions and exact compliance with boundary conditions. The FEONet approach infers coefficients, $\hat{\alpha}_k$, for constructing the linear combination $\sum \hat{\alpha}_k \phi_k$ to approximate PDE solutions. Thanks to the ability of FEM to handle boundary conditions, solutions predicted by FEONet also satisfy exact boundary conditions. Notably, its unique feature lies in solving parametric PDEs without relying on any paired input-output training data, a significant step forward in computational efficiency and application versatility. Another advantage of the FEONet is its applicability to singularly perturbed problems. By integrating the boundary layer element into the finite element space using the corrector basis function, we can establish an enriched basis scheme for utilization in the FEONet, allowing the model to capture sharp transitions accurately; see [32] for more details.

While significant progress has been made in scientific machine learning, a notable gap remains in the area of rigorous convergence analysis, which is essential for establishing the reliability of these novel methodologies [38, 15, 13]. Some analytical progress has been made with the Deep Ritz method [39, 37, 14], PINNs [46, 40], and Operator Network approaches [29, 28]. However, discrepancies between theoretical results and numerical experiment outcomes frequently arise, highlighting the need for more comprehensive and aligned studies in this area. In addition, to date, there has been a lack of convergence analysis for unsupervised operator networks such as the FEONet method. In this regard, a key part of our contribution is integrating the well-established FEM theory within the FEONet framework. This integration enables a more grounded and consistent convergence analysis, closely aligned with the outcomes of computational experiments. Our research focuses on this convergence analysis, bridging the gap between theoretical rigor and practical application. It highlights the potential of FEONet as a reliable tool for solving second-order elliptic PDEs, demonstrating its suitability for complex computational challenges.

The primary objective of the present paper is to perform a theoretical analysis of the FEONet, addressing the convergence of the method and deriving an error estimate. To be specific, we consider the

general second-order linear elliptic PDE of the form

$$-\operatorname{div}(\mathbf{a}(x)\nabla u) + \mathbf{b}(x) \cdot \nabla u + c(x)u = f(x) \quad \text{in } D, \quad (1.1)$$

$$u(x) = g(x) \quad \text{on } \partial D. \quad (1.2)$$

A similar type of theoretical analysis was performed in [27, 32]. Note, however, that the result in [27, 32] is restricted to the case of self-adjoint equations, and only a partial convergence result was obtained. In this paper, we will consider more general types of equations and investigate the comprehensive convergence properties of the FEONet. The main novelty of the present paper can be summarized as follows: In Section 3, we prove the convergence of the predicted solution by the FEONet. Our analysis is applicable to a general class of equations, which generalizes the previous results in [27, 32]. In the proof of convergence, we see that the convergence of the FEONet is influenced by the condition number of the finite element matrix, which is also confirmed by the numerical experiments. Subsequently in Section 4, unlike the previous papers where the parameters of baseline numerical methods were fixed, we identify the role of these parameters in overall convergence and conduct the complete theoretical analysis of the method. Additionally, in certain scenarios, we derive explicit error estimates for FEONet, utilizing a novel regularity theory developed for our method. Section 5 then presents numerical experiments that validate our theoretical findings. Furthermore, guided by the theoretical results obtained in the previous sections, we utilize the preconditioning techniques from the numerical analysis and confirm their significant impact on convergence and training efficiency.

In the next section, as a starting point, we briefly describe the idea of FEONet and demonstrate how it is trained and it predicts the solution within a suitable finite element setting. We also collect some results for the eigenvalue estimates of the finite element matrices which play an important role in the entire analysis. Additionally, we shall set up an analytic background for the theoretical analysis as well as the mathematical description of neural networks. Then the main sections of the paper proceed as described above, and some concluding remarks are made at the end of the paper.

2 Framework for finite element operator networks

In the equations (1.1)-(1.2), we shall assume that

$$\mathbf{a} \in L^\infty(D)^{d \times d}, \mathbf{b} \in W^{1,\infty}(D)^d, c \in L^\infty(D) \text{ and } f \in H^{-1}(D). \quad (2.1)$$

For the diffusion coefficients $\mathbf{a} = (a_{ij})$, we also assume that the uniform ellipticity condition holds: there exists a positive constant $\tilde{a} > 0$ such that

$$\sum_{i,j=1}^d a_{ij}(x)\xi_i\xi_j \geq \tilde{a} \sum_{i=1}^d \xi_i^2, \quad \forall \xi = (\xi_1, \dots, \xi_d) \in \mathbb{R}^d, \quad x \in \overline{D}. \quad (2.2)$$

For the well-posedness of the equation (1.1)-(1.2) we further assume that

$$c(x) - \frac{1}{2} \operatorname{div} \mathbf{b}(x) \geq 0, \quad x \in \overline{D}. \quad (2.3)$$

The corresponding weak formulation is defined as follows: find $u \in H_0^1(D)$ such that

$$B[u, v] := \int_D \mathbf{a}(x)\nabla u \cdot \nabla v \, dx + \int_D \mathbf{b}(x) \cdot \nabla u v \, dx + \int_D c(x)uv \, dx = \int_D f(x)v \, dx =: l(v) \quad \forall v \in H_0^1(D).$$

Thanks to the assumptions (2.1), (2.2) and (2.3), there holds for some constants c_0, c_1 and $c_2 > 0$ that

$$B[v, v] \geq c_0 \|v\|_{H^1(D)}, \quad |B[u, v]| \leq c_1 \|u\|_{H^1(D)} \|v\|_{H^1(D)}, \quad \text{and } |\ell(v)| \leq c_2 \|v\|_{H^1(D)}, \quad (2.4)$$

and the existence of a unique weak solution follows by the standard Lax–Milgram theory (see, e.g., [7]).

2.1 Finite element operator networks

In this section, we aim to describe FEONet, the main numerical scheme under consideration. This is a novel method introduced in [32], which utilizes the approximation power of deep neural networks in conjunction with the classical finite element approximation. More precisely, this method leverages the FEM for neural networks to learn the solution operator without any paired input-output training data to solve a wide range of parametric PDEs.

As a first step, let us define the finite element space, which will be used throughout the paper. Let \mathcal{G}_h be a shape-regular partition of a given physical domain \overline{D} , where h_E denotes the diameter of $E \in \mathcal{G}_h$ and $h = \max_{E \in \mathcal{G}_h} h_E$. We will also assume that there exists a positive constant $\gamma > 0$ independent of $h > 0$ such that $\max_{E \in \mathcal{G}_h} \frac{h_E}{\rho_E} \leq \gamma$, where ρ_E is the supremum of the diameters of inscribed balls for an element $E \in \mathcal{G}_h$. For a given partition \mathcal{G}_h , the finite element spaces are defined by $\mathbb{V}_h = \mathbb{V}(\mathcal{G}_h) := \{V \in C(\overline{D}) : V|_E \in \hat{\mathbb{P}}_V, E \in \mathcal{G}_h \text{ and } V|_{\partial D} = 0\}$, where $\hat{\mathbb{P}}_V \subset W^{1,\infty}(\hat{E})$ is a finite-dimensional subspace. Here we assume that \mathbb{V}_h has finite and locally supported basis; e.g., for each $h > 0$, there exists $N_h \in \mathbb{N}$ such that $\mathbb{V}_h = \text{span}\{\phi_1^h, \dots, \phi_{N_h}^h\}$ and for any basis function ϕ_i^h , $i = 1, \dots, N_h$, we have that if $\phi_i^h \neq 0$ on E for some $E \in \mathcal{G}_h$, then $\text{supp } \phi_j^h \subset \bigcup\{E' \in \mathcal{G}_h : E' \cap E \neq \emptyset\} =: S_E$. Furthermore, by shape regularity, it follows that $\exists C \in \mathbb{N}$ such that $|S_E| \leq C|E|$ for all $E \in \mathcal{G}_h$, where C can be chosen to be independent of h .

Next, we define our Galerkin approximation. We seek a discrete solution $u_h = \sum_{k=1}^{N_h} \alpha_k \phi_k \in \mathbb{V}_h$ satisfying

$$B[u_h, v_h] = \ell(v_h) \quad \text{for all } v_h \in \mathbb{V}_h. \quad (2.5)$$

If we write $S = (S_{ij})_{1 \leq i, j \leq N_h}$, $C = (C_{ij})_{1 \leq i, j \leq N_h} \in \mathbb{R}^{N_h \times N_h}$, $F = (F_j)_{1 \leq j \leq N_h} \in \mathbb{R}^{N_h}$ with

$$S_{ij} = \int_D [\mathbf{a}(x) \nabla \phi_i \cdot \nabla \phi_j + c(x) \phi_i \phi_j] dx, \quad C_{ij} = \int_D (\mathbf{b}(x) \cdot \nabla \phi_i) \phi_j dx, \quad F_j = \int_D f(x) \phi_j dx, \quad (2.6)$$

then the Galerkin approximation is equivalent to the linear algebraic equations

$$(S + C)\alpha = F, \quad \text{with } \alpha = (\alpha_k)_{1 \leq k \leq N_h} \in \mathbb{R}^{N_h}. \quad (2.7)$$

Now, let us introduce the finite element operator network, proposed in [32]. As mentioned earlier, the input of the FEONet can be any type of PDE data, for example, external force, variable coefficient, or boundary condition. Here, as a prototype example, we shall consider the networks whose input is an external forcing term. However, this can be extended to different types of input data in a straightforward manner. For each external forcing term f , instead of computing the coefficients using (2.7), we approximate the coefficients α using deep neural networks. For this purpose, we set the input of neural networks as external force f , parametrized by a random parameter ω for the (possibly high-dimensional) probability space $(\Omega, \mathcal{T}, \mathbb{P})$, assuming that Ω is compact. Typical examples are the Gaussian random fields or the random forcing terms defined by $f(x, \omega) = \omega_1 \sin(2\pi\omega_2 x) + \omega_3 \cos(2\pi\omega_4 x)$, where $\omega = (\omega_1, \omega_2, \omega_3, \omega_4)$ is i.i.d. distributed uniformly with $\omega_j \in [0, 1]$. In the present paper, we shall interpret $f(x, \omega)$ as a Bochner-type function defined on $D \times \Omega$, and we will assume the following throughout the paper.

Assumption 2.1. If we use the $(P\ell)$ -finite element approximation (piecewise polynomial function of degree less than or equal to $\ell \in \mathbb{N}$), we assume that $f(\cdot, \cdot) \in C(\Omega; H^{\ell-1}(D))$.

Once this input feature $\omega \in \Omega$ passes through the deep neural network, the coefficients $\{\hat{\alpha}_k\}$ are generated as an output. We then reconstruct the solution by

$$\hat{u}_h(x, \omega) = \sum_{k=1}^{N_h} \hat{\alpha}_k(\omega) \phi_k(x). \quad (2.8)$$

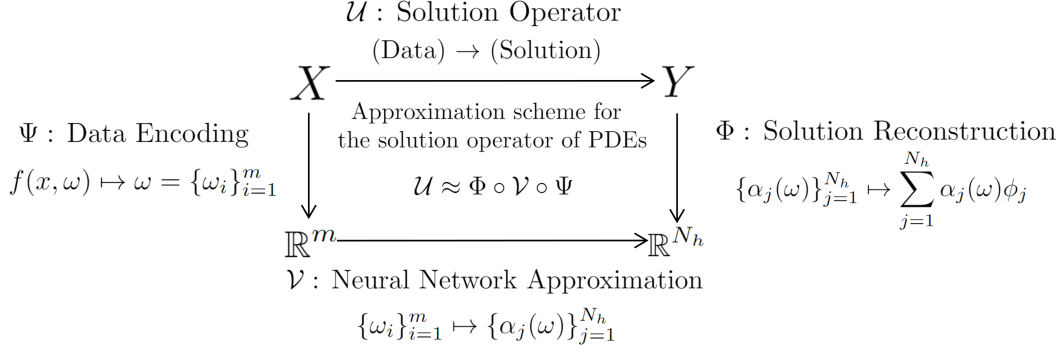


Figure 2: Schematic illustration of the decomposition of the FEONet into the encoder Ψ , approximator \mathcal{N} and reconstructor Φ .

For the training of the neural network, we set the population loss function as the L^2 -residual of the variational formulation (2.5):

$$\mathcal{L}(\alpha) = \mathbb{E}_{\omega \sim \mathbb{P}_\Omega} \left[\sum_{k=1}^{N_h} |B[\widehat{u}_h(x, \omega), \phi_k(x)] - \ell(\phi_k(x))|^2 \right]^{\frac{1}{2}}. \quad (2.9)$$

For the computational efficiency, when we actually compute the approximate solution, we deal with the empirical loss function which is the Monte–Carlo integration of (2.9):

$$\mathcal{L}^M(\alpha) = \frac{|\Omega|}{M} \sum_{j=1}^M \left[\sum_{k=1}^{N_h} |B[\widehat{u}_h(x, \omega_j), \phi_k(x)] - \ell(\phi_k(x))|^2 \right]^{\frac{1}{2}}, \quad (2.10)$$

where $\{\omega_j\}_{j=1}^M$ is a family of i.i.d. random samples selected from \mathbb{P}_Ω . For each training epoch, the parameters of the neural networks are updated toward minimizing the empirical loss \mathcal{L}^M , and then the given forcing term goes through the neural network again to predict more accurate coefficients. This process is repeated until we achieve a sufficiently small loss, and then we finally obtain the solution prediction (2.8). It is noteworthy that the FEONet only uses sample parameters randomly selected from Ω for the training, and hence it can be trained without any precomputed pairs of input-output training data. Moreover, as our method is based on the basis expansion (2.8), one can impose exact boundary values to the numerical solutions as we can do in the classical FEM. A schematic illustration of the decomposition of the FEONet is presented in Figure 2. Various numerical experiments on several benchmark problems can be found in [32], where we can confirm that our approach exhibited excellent performance, proving its versatility in terms of accuracy, generalization, and computational flexibility.

2.2 Estimations for eigenvalues of finite element matrices

As our FEONet approach is based on the finite element method, it is reasonable to expect that similar mathematical issues for the FEM may arise in the analysis of the FEONet. In fact, it will be made clear in the later analysis that the convergence of the FEONet is related to the condition number of the finite element matrices. In the classical FEM, adaptive or misshaped meshes may have an impact on the conditioning for the finite element approximation. This has a strong influence on the convergence of the approximation and the performance of iterative solvers for the resulting linear algebraic systems. For this reason, a condition number of finite element matrices has had a long history and has been of particular interest both theoretically and practically. Regarding this topic, a number of estimates of the largest and the smallest eigenvalues are available for both isotropic and anisotropic mesh geometries, and

some conditioning techniques have been established in various directions [7, 2, 1, 18, 25]. In the present paper, the discussion on such matters in full generality is out of scope, and we will only focus on the case of shape-regular partitions as we defined in the previous section. Therefore, the results we shall present here are not new and they can be regarded as special cases of the aforementioned results. However, for the sake of completeness, we will briefly present the derivation of the estimates in this section. These results will be used in the convergence analysis of the FEONet in later sections.

Let $A = S + C$ where the matrices S and C are defined in (2.6). For the estimation of the largest eigenvalue, we essentially follow the idea presented in [7]. Note here that since we are considering the shape-regular meshes, we may identify for all $E \in \mathcal{G}_h$ that $h_E \simeq h$, and $h^d \simeq N_h^{-1}$. By the inverse inequality (e.g. Eq. (1.5) in [7]), we have

$$\begin{aligned} \alpha^T A \alpha &= B[u_h, u_h] \lesssim \|u_h\|_{H^1(D)}^2 = \sum_{E \in \mathcal{G}_h} \|u_h\|_{H^1(E)}^2 \lesssim \sum_{E \in \mathcal{G}_h} h_E^{d-2} \|u_h\|_{L^\infty(E)}^2 \\ &\lesssim h^{d-2} \sum_{E \in \mathcal{G}_h} \sum_{\text{supp}(\phi_i) \cap E \neq \emptyset} \alpha_i^2 \lesssim h^{d-2} \alpha^T \alpha. \end{aligned}$$

This implies that

$$\lambda_{\max} \lesssim h^{d-2} \simeq N_h^{-1+\frac{2}{d}}, \quad (2.11)$$

where λ_{\max} denotes the largest eigenvalue of A . Next, we shall prove that N_h^{-1} is the lower bound of the smallest eigenvalue. By Poincaré's inequality and the inverse inequality, we have

$$\begin{aligned} \alpha^T A \alpha &= B[u_h, u_h] \gtrsim \|u_h\|_{H^1(D)}^2 \gtrsim \|u_h\|_{L^2(D)}^2 = \sum_{E \in \mathcal{G}_h} \|u_h\|_{L^2(E)}^2 \\ &\gtrsim h^d \sum_{E \in \mathcal{G}_h} \|u_h\|_{L^\infty(E)}^2 \gtrsim h^d \sum_{E \in \mathcal{G}_h} \sum_{\text{supp}(\phi_i) \cap E \neq \emptyset} \alpha_i^2 \gtrsim h^d \alpha^T \alpha. \end{aligned}$$

Therefore, we obtain

$$\lambda_{\min} \gtrsim h^d \simeq N_h^{-1}, \quad (2.12)$$

where λ_{\min} denotes the smallest eigenvalue of A .

Now let us make some comments on the estimate for the condition number of A . If the ℓ^2 norm is used to define the condition number, we can see that

$$\kappa(A) = \frac{\sigma_{\max}}{\sigma_{\min}}, \quad (2.13)$$

where σ_{\max} and σ_{\min} are maximal and minimal singular values of A respectively. Furthermore, if the matrix A is normal (i.e., it commutes its conjugate transpose), it is known that (see, e.g., [21]) the condition number $\kappa(A)$ of A satisfies

$$\kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}. \quad (2.14)$$

Therefore, if A is normal, together with the estimates (2.11) and (2.12), we may conclude that

$$\kappa(A) \lesssim N_h^{\frac{2}{d}} \simeq h^{-2}, \quad (2.15)$$

where N_h is the degree of freedom in the finite element formulation. On the other hand, in [16], the result for the lower bound of the condition number was obtained. More precisely, the authors proved the following estimate owing to the shape-regularity of the triangulation $\{\mathcal{G}_h\}_{h>0}$:

$$h^{-2} \lesssim \kappa(A). \quad (2.16)$$

From (2.16), (2.11) and (2.12), we can derive

$$\lambda_{\max} \lesssim \kappa(A)^{1-d/2} \quad \text{and} \quad \lambda_{\min}^{-1} \lesssim \kappa(A)^{d/2}, \quad (2.17)$$

which we will use later. The estimates obtained in this section will be utilized in the following sections to verify the relationship between the condition number and the convergence of the FEONet.

2.3 Feed forward neural networks

Next, we shall define a class of feed-forward ReLU neural networks, which we will consider throughout the paper. For each $L \in \mathbb{N}$, we denote an L -layer ReLU neural network by a function $f^L(x) : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$, defined recursively by

$$f^1(x) = W^1 x + b^1 \quad \text{and} \quad f^\ell(x) = W^\ell \sigma(f^{\ell-1}(x)) + b^\ell \quad \text{for } 2 \leq \ell \leq L, \quad (2.18)$$

where $W^\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ and $b^\ell \in \mathbb{R}^{n_\ell}$ is the weight matrix and the bias vector respectively for the ℓ -th layer. Here, $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is the ReLU activation function defined by $\sigma(x) = \max\{0, x\}$, and $\sigma(x)$ signifies the vector $(\sigma(x_1), \dots, \sigma(x_k))$ for $x = (x_1, \dots, x_k)$. We denote the architecture of a given network by the vector $\vec{n} = (n_0, \dots, n_L)$, the family of neural network parameters by $\theta := \theta_{\vec{n}} = \{(W^1, b^1), \dots, (W^L, b^L)\}$, and its realization as a function by $\mathcal{R}[\theta](x)$. For a given neural network architecture \vec{n} , the collection of all possible parameters is defined by

$$\Theta_{\vec{n}} = \left\{ \{(W^\ell, b^\ell)\}_{\ell=1}^L : W^\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}, b^\ell \in \mathbb{R}^{n_\ell} \right\}. \quad (2.19)$$

For two feed-forward neural networks θ_i , $i = 1, 2$ with architectures $\vec{n}_i = (n_0^{(i)}, \dots, n_{L_i}^{(i)})$, we shall write $\vec{n}_1 \subset \vec{n}_2$ if for any $\theta_1 \in \Theta_{\vec{n}_1}$, there exists $\theta_2 \in \Theta_{\vec{n}_2}$ satisfying $\mathcal{R}[\theta_1](x) = \mathcal{R}[\theta_2](x)$ for all $x \in \mathbb{R}^{n_0}$.

Now let us assume that there exists a sequence of neural network architectures $\{\vec{n}_n\}_{n \geq 1}$ satisfying $\vec{n}_n \subset \vec{n}_{n+1}$ for all $n \in \mathbb{N}$. We define the corresponding family of neural networks by

$$\mathcal{N}_n = \{\mathcal{R}[\theta] : \theta \in \Theta_{\vec{n}_n}\}. \quad (2.20)$$

It is straightforward to verify that $\mathcal{N}_n \subset \mathcal{N}_{n+1}$ for any $n \in \mathbb{N}$. We shall exploit the following theorem in the later analysis.

Theorem 2.2. *Let K be a compact set in \mathbb{R}^m and assume that $g \in C(K, \mathbb{R}^N)$. Then there holds*

$$\lim_{n \rightarrow \infty} \inf_{\hat{g} \in \mathcal{N}_n} \|\hat{g} - g\|_{C(K)} = 0. \quad (2.21)$$

It is called the universal approximation theorem and is known to hold for various scenarios. For instance, the original form of the universal approximation theorem for two-layer neural networks ($L = 2$) with an arbitrary number of nodes n exactly coincides with the above theorem [12, 22]. On the other hand, in the recent paper [26], the authors addressed an extension on the networks of arbitrary depth with bounded width. The authors assumed that the activation function σ is non-affine and continuously differentiable at some points, with non-vanishing derivatives at these points. For this case, the authors considered the class of neural networks with an arbitrary number of layers with $m + N + 2$ neurons for each, and derived (2.21).

2.4 Analytic framework

In this section, we shall describe the analytic framework which we will work within. Throughout the paper, given that the external force $f(x, \omega)$ parametrized by $\omega \in \Omega$, $u(x, \omega)$ signifies the corresponding solution, and $u_h(x, \omega)$ denotes the finite element approximation

$$u_h(x, \omega) = \sum_{k=1}^{N_h} \alpha_k^*(\omega) \phi_k(x), \quad (2.22)$$

where $\alpha^*(\omega) = \{\alpha_k^*(\omega)\}_{k=1}^{N_h}$ is the set of target finite element coefficients obtained from (2.7).

Next, we consider the approximation of the coefficients α^* by \mathbb{R}^{N_h} -valued ReLU neural networks. We seek for $\hat{\alpha}_n^{\mathcal{L}} : \Omega \rightarrow \mathbb{R}^{N_h}$ solving the minimization problem

$$\hat{\alpha}_n^{\mathcal{L}} = \arg \min_{\alpha \in \mathcal{N}_n} \mathcal{L}(\alpha), \quad (2.23)$$

where the minimization is over the neural networks class \mathcal{N}_n . Then we denote the associated solution by

$$u_{h,n}(x, \omega) = \sum_{k=1}^{N_h} (\hat{\alpha}_n^{\mathcal{L}})_k(\omega) \phi_k(x). \quad (2.24)$$

Finally, we shall consider the solution for the discrete minimization over the family of neural networks

$$\hat{\alpha}_{n,M}^{\mathcal{L}} = \arg \min_{\alpha \in \mathcal{N}_n} \mathcal{L}^M(\alpha), \quad (2.25)$$

where the empirical loss \mathcal{L}^M was defined in (2.10). In addition, we write the corresponding solution as

$$u_{h,n,M}(x, \omega) = \sum_{k=1}^{N_h} (\hat{\alpha}_{n,M}^{\mathcal{L}})_k(\omega) \phi_k(x). \quad (2.26)$$

In the present paper, we shall assume that the optimization error is negligible and that we can always find the exact minimizer for the optimization problems (2.23) and (2.25). Therefore, we shall regard $\hat{\alpha}_{n,M}^{\mathcal{L}}$ as the neural network approximation for the target coefficients α^* , and $u_{h,n,M}$ is our solution prediction computed by our FEONet scheme.

The main objective of this paper is to investigate the error $\|u - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}$. For this purpose, we split it into three parts:

$$u - u_{h,n,M} = \underbrace{(u - u_h)}_{\text{FEM error}} + \underbrace{(u_h - u_{h,n})}_{\text{approximation error}} + \underbrace{(u_{h,n} - u_{h,n,M})}_{\text{generalization error}} =: \text{(I)} + \text{(II)} + \text{(III)}. \quad (2.27)$$

The first error (I) arises from the finite element approximation and its mathematical analysis is fairly well-known. In particular, under suitable assumptions, it is known that (I) converges to 0 as $h \rightarrow 0$, and the convergence rate improves as we approximate the solution with higher-order polynomials assuming that the solution has suitable regularity properties. The second error (II) is referred to as the approximation error, occurring when the true coefficient α^* in (2.22) is approximated by a deep neural network. The final error (III) is known as the generalization error, which measures how well our predicted solution $u_{h,n,M}$ trained with M random samples generalizes for other samples that were not used in the training process; in other words, it is the error mainly caused by the approximation of \mathcal{L} by \mathcal{L}^M .

In the previous works [27, 32] where similar schemes were analyzed, all the convergence results were studied only when the index for the baseline numerical method (here denoted by $h > 0$) was fixed. Furthermore, the effect of $h > 0$ (or $N \in \mathbb{N}$ in [27] where the Legendre–Galerkin method was considered) to the errors (II) and (III) was not investigated in the previous analysis. In the present paper, however, we will derive a comprehensive error estimate, simultaneously examining the errors (I), (II), and (III) without fixing $h > 0$, and analyzing the role of the indices $h > 0$, n , $M \in \mathbb{N}$ in each error in (2.27).

As will be made clear in the later analysis, the convergence of the errors (II) and (III) is closely related to the condition number of the finite element matrices defined in (2.6). Based on the condition number estimates derived in Section 2.2, we will rigorously investigate the role of $h > 0$ on the approximation and generalization errors, which is a key novel part of the paper compared to the previous work [27, 32].

3 Convergence analysis for approximate solutions

We start with the following observation on the loss functions defined in (2.9) and (2.10). By the definition of the loss functions, the solution representation (2.8) and the bilinearity of $B[\cdot, \cdot]$, we note that

$$\begin{aligned}\mathcal{L}(\alpha) &= \int_{\Omega} \left[\sum_{i=1}^{N_h} \left| \sum_{k=1}^{N_h} \alpha_k(\omega) B[\phi_k(x), \phi_i(x)] - \ell(\phi_i(x)) \right|^2 \right]^{\frac{1}{2}} d\omega \\ &= \int_{\Omega} \left[\sum_{i=1}^{N_h} \left| (A\alpha(\omega))_i - (F(\omega))_i \right|^2 \right]^{\frac{1}{2}} d\omega = \|A\alpha(\omega) - F(\omega)\|_{L^1(\Omega)},\end{aligned}\tag{3.1}$$

where A and F are defined in (2.6). In the same manner, by the definition of population loss, we have

$$\mathcal{L}^M(\alpha) = \frac{|\Omega|}{M} \sum_{j=1}^M |A\alpha(\omega_j) - F(\omega_j)|,\tag{3.2}$$

where each ω_j is randomly chosen according to the distribution of \mathbb{P}_{Ω} . With the aid of the above observation, the loss functions \mathcal{L} and \mathcal{L}^M are represented in terms of the finite element matrices, and the analysis has now become a matter of the properties of these matrices. It is noteworthy that the matrix A contains the information of the given PDE and boundary conditions, and thus the characterization of A which can cover various PDE settings are of importance. In this perspective, the following lemma is useful in the analysis of the FEONet scheme regarding the structure of A , which is a direct consequence of the spectral theorem (see, e.g., Proposition 3.1 in [27]).

Lemma 3.1. *Suppose that T is a $N \times N$ symmetric and positive-definite matrix and let us denote minimum and maximum eigenvalues of T by λ_{\min} and λ_{\max} respectively. Then for any $x \in \mathbb{R}^N$, we have*

$$\lambda_{\min}|x| \leq |Tx| \leq \lambda_{\max}|x|.\tag{3.3}$$

In this section, we shall use the above lemma with the matrix $T = A^T A$, which is symmetric. Furthermore, by coercivity of $B[\cdot, \cdot]$, we find that A is positive-definite, and hence, $A^T A$ is also positive-definite. Therefore, even though A is not symmetric, we can apply Lemma 3.1 with $T = A^T A$. Based on this fact, in this section, we will derive estimates for the approximation error and the generalization error for the approximate coefficients $\|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)}$ and $\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}$ respectively. Before proceeding further, we introduce modified loss functions defined by

$$\mathcal{H}(\alpha) = \|A^T A\alpha(\omega) - A^T F(\omega)\|_{L^1(\Omega)} \quad \text{and} \quad \mathcal{H}^M(\alpha) = \frac{|\Omega|}{M} \sum_{i=1}^M |A^T A\alpha(\omega_i) - A^T F(\omega_i)|,$$

and their minimizers over the class of neural networks

$$\hat{\alpha}_n^{\mathcal{H}} = \arg \min_{\alpha \in \mathcal{N}_n} \mathcal{H}(\alpha) \quad \text{and} \quad \hat{\alpha}_{n,M}^{\mathcal{H}} = \arg \min_{\alpha \in \mathcal{N}_n} \mathcal{H}^M(\alpha).$$

3.1 Approximation error and Ceá's lemma

In this section, we aim to analyze the approximation error for the approximate coefficients. By Assumption 2.1, it is easy to verify that $F \in C(\Omega; \mathbb{R}^{N_h})$ and hence $\alpha^* = A^{-1}F \in C(\Omega; \mathbb{R}^{N_h})$. The precise statement for the approximation error estimate is encapsulated in the following theorem, which is a version of Ceá's lemma for the neural network approximation of the FEONet.

Theorem 3.2 (Approximation error). *Suppose that Assumption 2.1 holds and let $\kappa(A)$ denote the condition number of the finite element matrix $A = S + C$ defined in (2.6). For the finite element coefficients $\alpha^* \in C(\Omega; \mathbb{R}^{N_h})$ and the approximate coefficient $\hat{\alpha}_n^{\mathcal{L}} \in \mathcal{N}_n$, we have*

$$\|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} \lesssim \kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \quad (3.4)$$

Proof. By the triangle inequality, we can see that

$$\|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} \leq \|\alpha^* - \hat{\alpha}_n^{\mathcal{H}}\|_{L^1(\Omega)} + \|\hat{\alpha}_n^{\mathcal{H}} - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)}. \quad (3.5)$$

If we let ρ_{\max} and ρ_{\min} be maximum and minimum singular values of A respectively, (i.e., the square root of the eigenvalue of $A^T A$), from the repetitive application of Lemma 3.1 with $T = A^T A$, we have

$$\begin{aligned} \|\alpha^* - \hat{\alpha}_n^{\mathcal{H}}\|_{L^1(\Omega)} &\leq \frac{1}{(\rho_{\min})^2} (\|A^T A \alpha^* - A^T F\|_{L^1(\Omega)} + \|A^T A \hat{\alpha}_n^{\mathcal{H}} - A^T F\|_{L^1(\Omega)}) \\ &= \frac{1}{(\rho_{\min})^2} \mathcal{H}(\hat{\alpha}_n^{\mathcal{H}}) \leq \frac{1}{(\rho_{\min})^2} \inf_{\alpha \in \mathcal{N}_n} \mathcal{H}(\alpha) = \frac{1}{(\rho_{\min})^2} \inf_{\alpha \in \mathcal{N}_n} \|A^T A \alpha - A^T F\|_{L^1(\Omega)} \\ &\leq \frac{1}{(\rho_{\min})^2} \inf_{\alpha \in \mathcal{N}_n} (\|A^T A \alpha - A^T A \alpha^*\|_{L^1(\Omega)} + \|A^T A \alpha^* - A^T F\|_{L^1(\Omega)}) \\ &\leq \left(\frac{\rho_{\max}}{\rho_{\min}} \right)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} = \kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}, \end{aligned}$$

where we have used the minimality of $\hat{\alpha}_n^{\mathcal{H}}$ and the fact that $A^T A \alpha^* = A^T F$.

Next, let us estimate the second term on the right-hand side of (3.5). If we write the largest eigenvalue of A as λ_{\max} , by using the facts that $\|A^T\| = \|A\| = \rho_{\max}$ and $|\lambda_{\max}| \leq \rho_{\max}$ (see, e.g., [21]), we have

$$\begin{aligned} \|\hat{\alpha}_n^{\mathcal{H}} - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} &\leq \frac{1}{(\rho_{\min})^2} (\|A^T A \hat{\alpha}_n^{\mathcal{H}} - A^T F\|_{L^1(\Omega)} + \|A^T A \hat{\alpha}_n^{\mathcal{L}} - A^T F\|_{L^1(\Omega)}) \\ &\leq \left(\frac{\rho_{\max}}{\rho_{\min}} \right)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} + \frac{\|A^T\|}{(\rho_{\min})^2} \|A \hat{\alpha}_n^{\mathcal{L}} - F\|_{L^1(\Omega)} \\ &\leq \left(\frac{\rho_{\max}}{\rho_{\min}} \right)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} + \frac{\rho_{\max}}{(\rho_{\min})^2} \lambda_{\max} \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} \\ &\leq 2 \left(\frac{\rho_{\max}}{\rho_{\min}} \right)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} = 2\kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \end{aligned}$$

□

One interesting point is that the convergence of the approximate coefficient depends on the condition number of A . The FEONet uses a basis expansion similar to the classical FEM, but the essential difference is that the FEM solves the algebraic equation $A\alpha = F$, while the FEONet seeks to find a minimizer of $|A\alpha - F|$ over the class of neural networks. In the traditional FEM, iterative methods are commonly used to solve the equation $A\alpha = F$ (e.g., Krylov subspace methods, conjugate gradient methods) and the convergence of iterative methods depends on the condition number, which eventually affects the overall computation of the solution in the FEM. Therefore, it is natural to suspect whether the condition number also plays a role in computing the solution prediction of the FEONet, and the above theorem quantitatively confirms this relationship. As in the traditional FEM, we have seen that a high condition number also hinders the computation of solutions by the FEONet.

Another intriguing point is that, for a fixed finite element parameter $h > 0$, (and hence the condition number $\kappa(A)$), by the universal approximation theorem (2.21), we can prove that

$$\|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} \rightarrow 0 \quad \text{as } n \rightarrow \infty, \quad (3.6)$$

which is the desired convergence property of our approximate coefficients.

3.2 Generalization error

For the generalization error, we begin with the following definition so-called *Rademacher complexity*.

Definition 3.3. Let $\{X_i\}_{i=1}^M$ be a sequence of i.i.d. random variables. For a function class \mathcal{F} , we define the Rademacher complexity by

$$\mathcal{R}_M(\mathcal{F}) = \mathbb{E}_{\{X_i\}_{i=1}^M} \mathbb{E}_{\{\varepsilon_i\}_{i=1}^M} \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{M} \sum_{i=1}^M \varepsilon_i f(X_i) \right| \right],$$

where ε_i 's denote i.i.d. Bernoulli random variables, which means that $\mathbb{P}(\varepsilon_i = 1) = \mathbb{P}(\varepsilon_i = -1) = \frac{1}{2}$.

As we can see from the above definition, the Rademacher complexity is the expectation value of the maximum correlation between the vector $(f(X_1), \dots, f(X_M))$ and the random noise $(\varepsilon_1, \dots, \varepsilon_M)$, where the maximum is taken over the family of functions \mathcal{F} . This measures the capability of the class \mathcal{F} to fit random noise. For comprehensive information on the Rademacher complexity, see [17, 49]. The next theorem states that the difference between a population loss and an empirical loss can be bounded by the Rademacher complexity (see, for instance, Proposition 4.11 in [49]).

Theorem 3.4. Let \mathcal{F} be a class of function and $\{X_i\}_{i=1}^M$ be a sequence of i.i.d. random variables. Then we have the following inequality

$$\mathbb{E} \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{M} \sum_{j=1}^M f(X_j) - \mathbb{E}_{X \sim \mathbb{P}_\Omega} f(X) \right| \right] \leq 2\mathcal{R}_M(\mathcal{F}), \quad (3.7)$$

where the expectation is taken for the random variables $\{X_i\}_{i=1}^M$.

Next, we define the following function class of interest concerning the loss functions:

$$\mathcal{F}_n^{\mathcal{L}} := \{|A\alpha - F| : \alpha \in \mathcal{N}_n\}, \quad \mathcal{F}_n^{\mathcal{H}} := \{|A^T A\alpha - A^T F| : \alpha \in \mathcal{N}_n\}. \quad (3.8)$$

Now, we shall utilize Theorem 3.4 with both $\mathcal{F} = \mathcal{F}_n^{\mathcal{L}}$ and $\mathcal{F} = \mathcal{F}_n^{\mathcal{H}}$ to derive the estimate for the generalization error. The precise statement is presented in the following theorem.

Theorem 3.5 (Generalization error). *Let Assumption 2.1 holds and $\kappa(A)$ be the condition number of the finite element matrix $A = S + C$ defined in (2.6). Then we have*

$$\mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \lesssim \kappa(A)^{1+d/2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \kappa(A)^{d+2} R_M(\mathcal{F}_n^{\mathcal{H}}) + \kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \quad (3.9)$$

Proof. By the triangle inequality, we have

$$\begin{aligned} \mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] &\leq \mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_n^{\mathcal{H}}\|_{L^1(\Omega)}] + \mathbb{E} [\|\hat{\alpha}_n^{\mathcal{H}} - \hat{\alpha}_{n,M}^{\mathcal{H}}\|_{L^1(\Omega)}] + \mathbb{E} [\|\hat{\alpha}_{n,M}^{\mathcal{H}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \\ &=: \text{(I)} + \text{(II)} + \text{(III)} \end{aligned}$$

From the argument used in the proof of Theorem 3.2, we know that

$$\text{(I)} \lesssim \kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \quad (3.10)$$

Next, through the repeated applications of Lemma 3.1 and Theorem 3.4, we obtain

$$\begin{aligned}
(\text{II}) &\leq \frac{1}{\rho_{\min}^2} \mathbb{E} [\|A^T A \hat{\alpha}_n^{\mathcal{H}} - A^T F\|_{L^1(\Omega)} + \|A^T A \hat{\alpha}_{n,M}^{\mathcal{H}} - A^T F\|_{L^1(\Omega)}] \\
&= \frac{1}{\rho_{\min}^2} \mathbb{E} [\mathcal{H}(\hat{\alpha}_n^{\mathcal{H}}) + \mathcal{H}(\hat{\alpha}_{n,M}^{\mathcal{H}})] \lesssim \frac{1}{\rho_{\min}^2} \mathbb{E} [\mathcal{H}(\hat{\alpha}_{n,M}^{\mathcal{H}})] \\
&\lesssim \frac{1}{\rho_{\min}^2} (\mathbb{E} [\mathcal{H}(\hat{\alpha}_{n,M}^{\mathcal{H}}) - \mathcal{H}^M(\hat{\alpha}_{n,M}^{\mathcal{H}})] + \mathbb{E} [\mathcal{H}^M(\hat{\alpha}_{n,M}^{\mathcal{H}})]) \\
&\lesssim \frac{1}{\rho_{\min}^2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + \frac{2}{\rho_{\min}^2} \mathbb{E} [\mathcal{H}^M(\hat{\alpha}_n^{\mathcal{H}}) - \mathcal{H}(\hat{\alpha}_n^{\mathcal{H}})] + \frac{2}{\rho_{\min}^2} \mathbb{E} [\mathcal{H}(\hat{\alpha}_n^{\mathcal{H}})] \\
&\lesssim \frac{1}{\rho_{\min}^2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + \left(\frac{\rho_{\max}}{\rho_{\min}} \right)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)},
\end{aligned}$$

where we have used the argument used in the proof of Theorem 3.2 to obtain the last inequality. Finally,

$$(\text{III}) \leq \frac{1}{(\rho_{\min})^2} (\mathbb{E} [\|A^T A \hat{\alpha}_{n,M}^{\mathcal{H}} - A^T F\|_{L^1(\Omega)}] + \mathbb{E} [\|A^T A \hat{\alpha}_{n,M}^{\mathcal{L}} - A^T F\|_{L^1(\Omega)}]) =: \frac{1}{(\rho_{\min})^2} [(\text{I})' + (\text{II})'].$$

For the remaining terms, note that

$$\begin{aligned}
(\text{I})' &\leq \mathbb{E} [\mathcal{H}(\hat{\alpha}_{n,M}^{\mathcal{H}}) - \mathcal{H}^M(\hat{\alpha}_{n,M}^{\mathcal{H}})] + \mathbb{E} [\mathcal{H}^M(\hat{\alpha}_{n,M}^{\mathcal{H}})] \lesssim \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + \mathbb{E} [\mathcal{H}^M(\hat{\alpha}_n^{\mathcal{H}})] \\
&\lesssim \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + \mathbb{E} [\mathcal{H}^M(\hat{\alpha}_n^{\mathcal{H}}) - \mathcal{H}(\hat{\alpha}_n^{\mathcal{H}})] + \mathbb{E} [\mathcal{H}(\hat{\alpha}_n^{\mathcal{H}})] \lesssim \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + (\rho_{\max})^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)},
\end{aligned}$$

and subsequently,

$$\begin{aligned}
(\text{II})' &\leq \mathbb{E} [\|A^T \mathcal{L}(\hat{\alpha}_{n,M}^{\mathcal{L}})\|] \leq \|A^T\| (\mathbb{E} [\mathcal{L}(\hat{\alpha}_{n,M}^{\mathcal{L}}) - \mathcal{L}^M(\hat{\alpha}_{n,M}^{\mathcal{L}})] + \mathbb{E} [\mathcal{L}^M(\hat{\alpha}_{n,M}^{\mathcal{L}})]) \\
&\leq \|A^T\| (2\mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \mathbb{E} [\mathcal{L}^M(\hat{\alpha}_n^{\mathcal{L}}) - \mathcal{L}(\hat{\alpha}_n^{\mathcal{L}})] + \mathbb{E} [\mathcal{L}(\hat{\alpha}_n^{\mathcal{L}})]) \\
&\lesssim \|A^T\| \left(4\mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \lambda_{\max} \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} \right).
\end{aligned}$$

Again, by the facts that $\|A^T\| = \|A\| = \rho_{\max}$ and $|\lambda_{\max}| \leq \rho_{\max}$ together with the above estimates,

$$(\text{III}) \lesssim \kappa(A)^{d+2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{H}}) + \kappa(A)^{1+d/2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \kappa(A)^2 \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}.$$

By combining the above estimates for (I), (II) and (III), together with the fact $|\lambda_{\max}| \leq \rho_{\max}$, we can derive the desired estimate. \square

If the finite element parameter $h > 0$ is fixed, along with ρ_{\min} , ρ_{\max} , and $\kappa(A)$, then, according to the universal approximation property, we know that the last term on the right-hand side of (3.9) converges to zero. For the first term and the second term, we may assume that the Rademacher complexities converge to zero as $M \rightarrow \infty$; this is a common assumption in statistical learning theory, and it indeed holds for several function families. See, for example, [41, 14, 37, 20], where this issue was addressed. Therefore, in this case, we can show that for fixed $h > 0$, the generalization error converges to zero as $n, M \rightarrow \infty$. More precisely, we have

$$\mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \rightarrow 0 \quad \text{as } n, M \rightarrow \infty. \quad (3.11)$$

3.3 Convergence of approximate solutions

Based on the convergence of coefficients (3.6) and (3.11), we shall prove the convergence of the FEONet prediction to the finite element approximation, which is encapsulated in the following theorem.

Theorem 3.6. Suppose Assumption 2.1 holds, and assume that for all $n \in \mathbb{N}$, $\mathcal{R}_M(\mathcal{F}_n^\mathcal{L})$ and $\mathcal{R}_M(\mathcal{F}_n^\mathcal{H})$ converge to 0 as $M \rightarrow \infty$. Then we have

$$\lim_{n \rightarrow \infty} \lim_{M \rightarrow \infty} \mathbb{E} [\|u_h - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}] = 0, \quad (3.12)$$

where the expectation is to take over the random sampling $\omega_j \sim \mathbb{P}_\Omega$.

Proof. From the definition (2.22) and (2.26), we have that

$$\begin{aligned} \|u_h - u_{h,n,M}\|_{L^1(\Omega; L^2(D))} &= \int_\Omega \left\| \sum_{j=1}^{N_h} (\alpha_j^* - (\hat{\alpha}_{n,M}^\mathcal{L})_j) \phi_j \right\|_{L^2(D)} d\omega \lesssim \int_\Omega \left(\sum_{j=1}^{N_h} (\alpha_j^* - (\hat{\alpha}_{n,M}^\mathcal{L})_j) \|\phi_j\|_{L^2(D)} \right) d\omega \\ &\lesssim \max_{1 \leq i \leq N_h} \|\phi_j\|_{L^2(D)} \int_\Omega \left(\sum_{j=1}^{N_h} |\alpha_j^* - (\hat{\alpha}_{n,M}^\mathcal{L})_j| \right) d\omega \\ &\lesssim \max_{1 \leq i \leq N_h} \|\phi_j\|_{L^2(D)} N_h^{1/2} \|\alpha^* - \hat{\alpha}_{n,M}^\mathcal{L}\|_{L^1(\Omega)}. \end{aligned} \quad (3.13)$$

Therefore, for given $h > 0$, by (3.6) and (3.11), we conclude that as $n, M \rightarrow \infty$,

$$\mathbb{E} [\|u_h - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}] \lesssim \mathbb{E} [\|\alpha^* - \hat{\alpha}_{n,M}^\mathcal{L}\|_{L^1(\Omega)}] \lesssim \mathbb{E} [\|\alpha^* - \hat{\alpha}_n^\mathcal{L}\|_{L^1(\Omega)}] + \mathbb{E} [\|\hat{\alpha}_n^\mathcal{L} - \hat{\alpha}_{n,M}^\mathcal{L}\|_{L^1(\Omega)}] \rightarrow 0.$$

□

4 Error estimates for approximate solutions

In the previous section, we considered the general class of second-order elliptic equations (1.1)-(1.2) and proved that the error $\|u_h - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}$ goes to zero as $n, M \rightarrow \infty$. On the other hand, in this section, we will deal with some particular cases where we can derive an explicit error estimate.

First, we shall consider the case of self-adjoint PDEs, i.e., the case when $\mathbf{b}(x) = 0$ for all $x \in D$. In this case, we can significantly simplify the proof in Section 3 by avoiding the use of the auxiliary loss function \mathcal{H} . Furthermore, by using the condition number estimates of finite element matrices, we can identify the role of the finite element parameter $h > 0$ in the entire convergence and obtain the explicit error bound. Secondly, we will introduce a function space called the Barron space, which is the family of functions endowed with a quantity that can control the approximation and generalization errors by adopting a particular machine learning model. We will provide a sufficient condition for our target function to be contained in the Barron space, and derive the explicit error bounds which identify the role of the parameters $h > 0, n, M \in \mathbb{N}$ in the overall convergence of the FEONet.

4.1 Self-adjoint PDEs

Henceforth, in this section, let us assume that $\mathbf{b}(x) = 0$ for any $x \in D$ in (1.1), so that the equations under consideration become self-adjoint:

$$-\operatorname{div}(\mathbf{a}(x)\nabla u) + c(x)u = f(x) \quad \text{in } D, \quad (4.1)$$

$$u(x) = 0 \quad \text{on } \partial D. \quad (4.2)$$

We shall adjust the analytical setting discussed in Section 2 accordingly, so all the notations are now for the equation (4.1)-(4.2). In this case, the finite element matrix A is just S (instead of $S + C$ as before), which is symmetric and positive-definite. Therefore, we can directly apply Lemma 3.1 with $T = A = S$, which allows us to avoid the use of the auxiliary loss function \mathcal{H} . As we can see from the following theorems, since we only work the original loss function \mathcal{L} , the main assertions and the proofs can be

significantly simplified. Again, let us denote the condition number of the finite element matrix $A = S$ by $\kappa(A)$, and the largest eigenvalue and the smallest eigenvalue by λ_{\max} and λ_{\min} respectively. Note that since A is symmetric (and hence normal) and positive-definite, we can write $\kappa(A) = \lambda_{\max}/\lambda_{\min}$.

Theorem 4.1 (Approximation error). *Suppose that Assumption 2.1 holds. For the finite element coefficients $\alpha^* \in C(\Omega; \mathbb{R}^{N_h})$ and the approximate coefficient $\hat{\alpha}_n^{\mathcal{L}} \in \mathcal{N}_n$, we have*

$$\|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} \leq \kappa(A) \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \quad (4.3)$$

Proof. By using Lemma 3.1 with $T = A = S$, it follows that

$$\begin{aligned} \|\alpha^* - \hat{\alpha}_n^{\mathcal{L}}\|_{L^1(\Omega)} &\leq \frac{1}{\lambda_{\min}} (\|A\alpha^* - F\|_{L^1(\Omega)} + \|A\hat{\alpha}_n^{\mathcal{L}} - F\|_{L^1(\Omega)}) \\ &= \frac{1}{\lambda_{\min}} \mathcal{L}(\hat{\alpha}_n^{\mathcal{L}}) \leq \frac{1}{\lambda_{\min}} \inf_{\alpha \in \mathcal{N}_n} \mathcal{L}(\alpha) = \frac{1}{\lambda_{\min}} \inf_{\alpha \in \mathcal{N}_n} \|A\alpha - F\|_{L^1(\Omega)} \\ &\leq \frac{1}{\lambda_{\min}} \inf_{\alpha \in \mathcal{N}_n} (\|A\alpha - A\alpha^*\|_{L^1(\Omega)} + \|A\alpha^* - F\|_{L^1(\Omega)}) \\ &\leq \frac{\lambda_{\max}}{\lambda_{\min}} \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} = \kappa(A) \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \end{aligned}$$

□

Theorem 4.2 (Generalization error). *If Assumption 2.1 holds, we have*

$$\mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \lesssim \kappa(A)^{d/2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \kappa(A) \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \quad (4.4)$$

Proof. By the use of Lemma 3.1 and Theorem 3.4, we have

$$\begin{aligned} \mathbb{E} [\|\hat{\alpha}_n^{\mathcal{L}} - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] &\leq \frac{1}{\lambda_{\min}} \mathbb{E} [\|A\hat{\alpha}_n^{\mathcal{L}} - F\|_{L^1(\Omega)} + \|A\hat{\alpha}_{n,M}^{\mathcal{L}} - F\|_{L^1(\Omega)}] \\ &= \frac{1}{\lambda_{\min}} \mathbb{E} [\mathcal{L}(\hat{\alpha}_n^{\mathcal{L}}) + \mathcal{L}(\hat{\alpha}_{n,M}^{\mathcal{L}})] \leq \frac{1}{\lambda_{\min}} \mathbb{E} [\mathcal{L}(\hat{\alpha}_{n,M}^{\mathcal{L}})] \\ &\lesssim \frac{1}{\lambda_{\min}} (\mathbb{E} [\mathcal{L}(\hat{\alpha}_{n,M}^{\mathcal{L}}) - \mathcal{L}^M(\hat{\alpha}_{n,M}^{\mathcal{L}})] + \mathbb{E} [\mathcal{L}^M(\hat{\alpha}_{n,M}^{\mathcal{L}})]) \\ &\lesssim \frac{1}{\lambda_{\min}} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \frac{1}{\lambda_{\min}} \mathbb{E} [\mathcal{L}^M(\hat{\alpha}_n^{\mathcal{L}}) - \mathcal{L}(\hat{\alpha}_n^{\mathcal{L}})] + \frac{1}{\lambda_{\min}} \mathbb{E} [\mathcal{L}(\hat{\alpha}_n^{\mathcal{L}})] \\ &\lesssim \frac{1}{\lambda_{\min}} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) + \frac{\lambda_{\max}}{\lambda_{\min}} \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)}. \end{aligned}$$

Therefore, from (2.12) and (2.16), we obtain the desired inequality. □

Now, from Theorem 4.1 and Theorem 4.2, we have the following estimates:

$$\mathbb{E} [\|\alpha^* - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \lesssim \kappa(A) \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} + \kappa(A)^{d/2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}). \quad (4.5)$$

The above estimate plays a pivotal role in the error analysis. For example, from the above inequality together with the classical finite element theory, we can estimate an error between the true solution u and the solution prediction $u_{h,n,M}$ by the FEONet. More precisely, for the $(P\ell)$ -finite element approximation (piecewise polynomial function degree less than or equal to $\ell \in \mathbb{N}$), by the optimal error estimate of finite element approximation and the elliptic regularity theory, it follows that

$$\|u - u_h\|_{L^1(\Omega; L^2(D))} \lesssim h^{\ell+1} \int_{\Omega} |u(\omega)|_{H^{\ell+1}(D)} d\omega \lesssim h^{\ell+1} \int_{\Omega} \|f(\omega)\|_{H^{\ell-1}(D)} d\omega = h^{\ell+1} \|f\|_{L^1(\Omega; H^{\ell-1}(D))}.$$

Furthermore, for the finite element basis function ϕ_j^h on the shape-regular partition, it is straightforward to verify that $\|\phi_j^h\|_{L^2(D)} \approx h^{d/2}$. Therefore, as we did in (3.13), we can obtain the following estimate:

$$\begin{aligned} \mathbb{E} [\|u - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}] &\lesssim h^{\ell+1} + \max_{1 \leq i \leq N_h} \|\phi_i\|_{L^2(D)} N_h^{1/2} \mathbb{E} [\|\alpha^* - \hat{\alpha}_{n,M}^{\mathcal{L}}\|_{L^1(\Omega)}] \\ &\lesssim h^{\ell+1} + \kappa(A) \inf_{\alpha \in \mathcal{N}_n} \|\alpha - \alpha^*\|_{L^1(\Omega)} + \kappa(A)^{d/2} \mathcal{R}_M(\mathcal{F}_n^{\mathcal{L}}) \end{aligned} \quad (4.6)$$

Unlike the convergence obtained in Theorem 3.6 where the error $u_h - u_{h,n,M}$ with fixed $h > 0$ was addressed, here we do not fix $h > 0$ and investigate the total error $u - u_{h,n,M}$ (instead of $u_h - u_{h,n,M}$), identifying the role of $h > 0$ in the entire convergence. It is noteworthy from the estimate (4.6), that the choice of small $h > 0$ may not always guarantee a small error, and we need to choose a suitable $h > 0$ to minimize the error for the predicted solution. In the next chapter, with some regularity assumption, we will further estimate the right-hand side of (4.6) in terms of $n, M \in \mathbb{N}$ to obtain the complete error estimate, which is the main goal of this paper.

4.2 The Barron Space

In this section, we shall introduce a certain class of functions called the Barron space, to honor the seminal work of Barron concerning the mathematical analysis of a class of two-layer neural networks. It has been studied that this type of function can be well approximated by two-layer neural networks with dimension-independent convergence rates with respect to the width [4, 14]. We start with the so-called spectral Barron space which was Barron's original approach [47, 48, 37]. For an integrable function f defined on $\Omega \subset \mathbb{R}^m$, we define the quantity

$$\gamma_s(f_e) = \int_{\mathbb{R}^m} (1 + |\xi|)^s |\hat{f}_e(\xi)| d\xi, \quad (4.7)$$

where $f_e \in L^1(\mathbb{R}^m)$ is an extension of f to \mathbb{R}^m and \hat{f}_e is the Fourier transform of f_e . We call the space of functions with bounded $\gamma_s(\cdot)$ as the Barron space. More precisely, we shall define

$$\mathcal{B}^s(\Omega) := \{f : \Omega \rightarrow \mathbb{R} : \gamma_s(f_e) < \infty \text{ for some extension of } f_e \text{ of } f\},$$

which is equipped with the norm

$$\|f\|_{\mathcal{B}^s(\Omega)} := \inf_{f_e|_{\Omega}=f} \gamma(f_e),$$

where the infimum is taken over all extensions $f_e \in L^1(\mathbb{R}^m)$. Then one can show that $\mathcal{B}^s(\Omega)$ is continuously embedded in $H^s(\Omega)$ provided that Ω is bounded [47]. There are some results providing the characterization of the Barron Space. For example, a sufficiently smooth function with compact support is contained in the Barron space, which is presented in the following lemma (see, e.g., [19]).

Lemma 4.3. *If $g \in C_0^\beta(\mathbb{R}^m)$ with $|\beta| > m/2 + s$ for some $s \in \mathbb{N}$, then $g \in \mathcal{B}^s(\mathbb{R}^m)$ in the sense that*

$$\gamma_s(g)^2 \lesssim \int_{\mathbb{R}^m} (|g|^2 + |\partial^\beta g|^2) dx < \infty. \quad (4.8)$$

More characterizations and properties can be found in various papers including [4, 5, 47]. Note that, if the function is defined in a bounded domain Ω rather than the whole space \mathbb{R}^m , we need to extend the functions from Ω to \mathbb{R}^m . This can be done by a continuous extension argument such as the Whitney extension theorem which is encapsulated in the following theorem (see, e.g., [19]).

Lemma 4.4. *If $g \in C^\beta(\overline{\Omega})$, then for a closed set $\Gamma \subset \mathbb{R}^m$ satisfying $\Omega \subset \subset \Gamma$, there exists an extension $g_e \in C_0^\beta(\mathbb{R}^m)$ such that $g = g_e$ in $\overline{\Omega}$ and $g_e = 0$ in $\mathbb{R}^m \setminus \Gamma$.*

In modern machine learning theory, the Barron space is sometimes defined in a different way, using a probabilistic integral representation [14, 8]. More precisely, let us consider a function $g : \Omega \rightarrow \mathbb{R}$ with the following integral representation

$$g(\omega) = \int_{\mathbb{R} \times \mathbb{R}^m \times \mathbb{R}} a\sigma(b \cdot \omega + c)\rho(da, db, dc), \quad (4.9)$$

where ρ is a probability distribution on $\mathbb{R} \times \mathbb{R}^m \times \mathbb{R}$ and σ is an activation function. Then we define the norm of such function as

$$\|g\|_{\mathcal{W}^s(\Omega)} = \inf_{\rho} (\mathbb{E}_{\rho}[|a|^s(\|b\|_1 + |c|)^s])^{1/s},$$

where the infimum is taken over all ρ with the representation (4.9). The probabilistic Barron space $\mathcal{W}^s(\Omega)$ is defined as the class of continuous functions which can be represented by (4.9) with finite Barron norm $\|\cdot\|_{\mathcal{W}^s(\Omega)}$. This is less explicit than the Fourier-based characterization, but it is known to contain more functions that can be efficiently approximated by two-layer neural networks [43, 44].

The regularity we will mainly concern in this paper is $\mathcal{B}^2(\Omega)$, and hence, we shall denote $\mathcal{B}(\Omega) = \mathcal{B}^2(\Omega)$ henceforth. In several papers, it was shown that $\mathcal{B}^{s+1}(\Omega)$ is continuously embedded into $\mathcal{W}^s(\Omega)$ for all $s \in \mathbb{N}$ [4, 51], and $\mathcal{W}^1(\Omega) = \mathcal{W}^2(\Omega) = \mathcal{W}^3(\Omega) = \dots = \mathcal{W}^\infty(\Omega)$ provided that σ is the ReLU activation function [14]. Therefore we will use the single notation $\mathcal{W}(\Omega)$ to denote the probabilistic Barron space. From the above properties, it follows that $\mathcal{B}(\Omega) \hookrightarrow \mathcal{W}(\Omega)$.

We begin with the following regularity result which identifies a sufficient condition for our target coefficient α^* to be contained in the Barron space. To do this, we need to assume the following.

Assumption 4.5. For a compact set $\Omega \subset \mathbb{R}^m$, $f(x, \omega) \in L^1(D; C^\beta(\overline{\Omega}))$ with $|\beta| > m/2 + 2$.

Proposition 4.6. Suppose that Assumption 4.5 holds. Then each target coefficient (finite element coefficient) α_j^* is contained in $\mathcal{B}(\Omega)$ for $j = 1, \dots, N_h$ with the estimate

$$\|\alpha_j^*\|_{\mathcal{B}(\Omega)} \lesssim \kappa(A)^{d/2}.$$

Proof. By recalling the definition of the finite element coefficients α^* , for each $j \in N_h$

$$\alpha_j^*(\omega) = \tilde{a}_1^j F_1(\omega) + \dots + \tilde{a}_{N_h}^j F_{N_h}(\omega) = \int_D f(x, \omega) [\tilde{a}_1^j \phi_1(x) + \dots + \tilde{a}_{N_h}^j \phi_{N_h}(x)] dx =: \int_D f(x, \omega) B_j(x) dx,$$

where $(\tilde{a}_1^j, \dots, \tilde{a}_{N_h}^j)$ denotes the j -th row of A^{-1} . Since the largest component of the inverse of A is bounded above by λ_{\min}^{-1} (see, e.g., [21]), from (2.17), we see that $|B_j| \lesssim \kappa(A)^{d/2}$.

Next, by Assumption 4.5 and Lemma 4.4, there exists an extension $f_{e_x} \in C_0^\beta(\mathbb{R}^m)$ for each $x \in D$. We then define an extension of α_j^* by

$$\alpha_{j,e}^*(\omega) = \int_D f_{e_x}(x, \omega) B_j(x) dx.$$

By Fubini's theorem, we have

$$\begin{aligned} \widehat{\alpha_{j,e}^*}(\xi) &= \int_{\mathbb{R}^m} \left[\int_D f_{e_x}(x, \omega) B_j(x) dx \right] e^{-i\xi \cdot \omega} d\omega \\ &\lesssim \kappa(A)^{d/2} \int_D \int_{\mathbb{R}^m} f_{e_x}(x, \omega) e^{-i\xi \cdot \omega} d\omega dx = \kappa(A)^{d/2} \int_D \widehat{f_{e_x}}(x, \xi) dx. \end{aligned}$$

Again, by Fubini's theorem together with (4.8) and the continuity of the extension e_x , it follows that

$$\begin{aligned} \gamma_s(\alpha_{j,e}^*) &= \int_{\mathbb{R}^m} (1 + |\xi|)^s |\widehat{\alpha_{j,e}^*}(\xi)| d\xi \lesssim \kappa(A)^{d/2} \int_{\mathbb{R}^m} (1 + |\xi|)^s \int_D |\widehat{f_{e_x}}(x, \xi)| dx d\xi \\ &\lesssim \kappa(A)^{d/2} \int_D \int_{\mathbb{R}^m} (1 + |\xi|)^s |\widehat{f_{e_x}}(x, \xi)| d\xi dx \lesssim \kappa(A)^{d/2} \int_D \gamma_s(f_{e_x}) dx \lesssim \kappa(A)^{d/2} \int_D \|f\|_{C^\beta(\overline{\Omega})} dx. \end{aligned}$$

From Assumption 4.5, we can conclude that $\alpha_j^* \in \mathcal{B}(\Omega)$ for all $j = 1, \dots, N_h$. \square

Based on Proposition 4.6, henceforth, we will assume that our target coefficient satisfies $\alpha_j^* \in B(\Omega)$ for all $1 \leq j \leq N_h$. Next, let us define an ansatz space for approximation. In the previous sections, we used the general class of feed-forward neural networks as approximators. On the other hand, here we use a set of two-layer ReLU neural networks, which is known to effectively approximate the function of Barron type. Specifically, the class of feed-forward neural networks \mathcal{N}_n defined in Section 2.3 now denotes the family of two-layer ReLU networks from Ω into \mathbb{R}^{N_h} , which contains n neurons in the hidden layer. Also, we shall use the following class of scalar-valued neural networks:

$$\mathcal{N}_{n,2} := \left\{ \alpha(\omega; \theta) = \frac{1}{n} \sum_{j=1}^n a_j \sigma(b_j \cdot \omega + c_j) : \theta = (a_j, b_j, c_j)_{j=1}^n, a_j, c_j \in \mathbb{R}, b_j \in \mathbb{R}^m \text{ for all } 1 \leq j \leq n \right\}.$$

Furthermore, for a two-layer neural network with the parameter $\theta = (a_j, b_j, c_j)_{j=1}^n$, we define the so-called path norm by

$$\|\theta\|_{\mathcal{P}} := \frac{1}{n} \sum_{j=1}^n |a_j| (\|b_j\|_1 + |c_j|).$$

Then we define the main ansatz class in this section which consists of the two-layer ReLU neural networks with the parameters whose path norms are bounded by the Barron norm of the target coefficients:

$$\mathcal{N}_{n,2}^* := \left\{ \alpha(\omega; \theta) \in \mathcal{N}_n : \|\theta\|_{\mathcal{P}} \leq 2 \max_{1 \leq j \leq N_h} \|\alpha_j^*\|_{\mathcal{W}(\Omega)} \right\}.$$

It is known that if the target function is contained in the Barron space, it can be well approximated by two-layer neural networks without the curse of dimensionality which is presented in the following theorem (see, for example, [4, 14]).

Theorem 4.7. *Let $g \in \mathcal{W}(\Omega)$ and $n \in \mathbb{N}$. Then there exists a two-layer ReLU neural network $g_n(\cdot; \theta) \in \mathcal{N}_{n,2}$ with $\|\theta\|_{\mathcal{P}} \leq 2\|g\|_{\mathcal{W}(\Omega)}$ satisfying*

$$\|g(\cdot) - g_n(\cdot; \theta)\|_{L^2(\Omega)}^2 \leq \frac{3\|g\|_{\mathcal{W}(\Omega)}^2}{n}.$$

Note that the above theorem is for the scalar-valued neural networks. Let us describe the way to apply Theorem 4.7 to our case, where the vector-valued neural networks are considered. For the target coefficient $\alpha^*(\omega) = (\alpha_i^*(\omega))_{i=1}^{N_h}$ with $\alpha_i^* \in \mathcal{W}(\Omega)$, by Theorem 4.7, there exists two-layer neural networks

$$\alpha_{i,n}^*(\omega) = \frac{1}{n} \sum_{j=1}^n a_j^i \sigma(b_j^i \cdot \omega + c_j^i) \in \mathcal{N}_{n,2}^*$$

for each $i = 1, \dots, N_h$, satisfying

$$\|\alpha_i^* - \alpha_{i,n}^*\|_{L^2(\Omega)}^2 \leq \frac{3\|\alpha_i^*\|_{\mathcal{W}(\Omega)}^2}{n}.$$

Now we define a (vector-valued) two-layer ReLU neural network $\hat{g} : \Omega \rightarrow \mathbb{R}^{N_h}$ with nN_h nodes such that the weight matrix and the bias vector of the hidden layer is defined by

$$W^1 = (b_1^1, \dots, b_n^1, \dots, b_1^{N_h}, \dots, b_n^{N_h})^T \in \mathbb{R}^{nN_h \times n} \quad \text{and} \quad b^1 = (c_1^1, \dots, c_n^1, \dots, c_1^{N_h}, \dots, c_n^{N_h})^T \in \mathbb{R}^{nN_h}.$$

The weight matrix of the second layer is defined by

$$W^2 = \begin{pmatrix} a_1^1 \dots a_n^1 & 0 \dots 0 & 0 \dots 0 & \dots & 0 \dots 0 \\ 0 \dots 0 & a_1^2 \dots a_n^2 & 0 \dots 0 & \dots & 0 \dots 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \dots 0 & 0 \dots 0 & 0 \dots 0 & \dots & a_1^{N_h} \dots a_n^{N_h} \end{pmatrix}.$$

Then we have

$$\begin{aligned} \inf_{g \in \mathcal{N}_{nN_h}} \|\alpha^* - g\|_{L^1(\Omega)} &\lesssim \|\alpha^* - \widehat{g}\|_{L^2(\Omega)} = \left(\|\alpha_1^* - \alpha_{1,n}^*\|_{L^2(\Omega)}^2 + \cdots + \|\alpha_{N_h}^* - \alpha_{N_h,n}^*\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}} \\ &\lesssim \frac{1}{\sqrt{n}} \left(\|\alpha_1^*\|_{\mathcal{B}(\Omega)}^2 + \cdots + \|\alpha_{N_h}^*\|_{\mathcal{B}(\Omega)}^2 \right)^{\frac{1}{2}} \lesssim \frac{\kappa(A)^{3d/4}}{\sqrt{n}}. \end{aligned}$$

Therefore, we obtain the following result.

Proposition 4.8. *Suppose that Assumption 4.5 holds. Then we have*

$$\inf_{\alpha \in \mathcal{N}_n} \|\alpha^* - \alpha\|_{L^1(\Omega)} \lesssim \frac{\kappa(A)^d}{\sqrt{n}}. \quad (4.10)$$

Next, we shall investigate the generalization error. To do this, we will use the following result [14, 42].

Theorem 4.9. *For the function class $\mathcal{F}_{Q,1} = \{g(\cdot; \theta) \in \mathcal{N}_{n,2} : \|\theta\|_{\mathcal{P}} \leq Q\}$ and $\mathcal{F}_{Q,2} = \{g \in \mathcal{W} : \|g\|_{\mathcal{W}} \leq Q\}$, we have*

$$\mathcal{R}_M(\mathcal{F}_{Q,j}) \leq 2Q \sqrt{\frac{2 \log(2m+2)}{M}} \quad \text{for } j = 1, 2. \quad (4.11)$$

Let us write $A = (a_{ij})$ and $a_{\max} = \max_{i,j} |a_{ij}|$. Since the finite element matrix A is symmetric and positive-definite, there exists some $k \in \mathbb{N}$ such that $|a_{ij}| \leq a_{kk}$ for any $i, j = 1, 2, \dots, N_h$ (see, e.g., [21]). Therefore, from the estimate (2.17), it follows that $a_{\max} \leq |a_{kk}| = e_k^T A e_k \leq \lambda_{\max} \lesssim \kappa(A)^{1-d/2}$, where e_k denotes the k -th standard basis. In order to apply Theorem 4.9 to our case, let us first define the following function classes:

$$\begin{aligned} \mathcal{F}_n &:= \{|A\alpha - F| : \alpha = (\alpha_j)_{j=1}^{N_h}, \alpha_j \in \mathcal{N}_{n,2}^* \text{ for } j = 1, \dots, N_h\}, \\ \mathcal{F}_{n,1} &:= \{|A\alpha| : \alpha = (\alpha_j)_{j=1}^{N_h}, \alpha_j \in \mathcal{N}_{n,2}^* \text{ for } j = 1, \dots, N_h\}, \\ \mathcal{F}_{n,1}^{i,j} &:= \{|a_{ij}g| : g \in \mathcal{N}_{n,2}^*\}, \quad \mathcal{F}_{n,1}^{\max} := \{a_{\max}g : g \in \mathcal{N}_{n,2}^*\}. \end{aligned}$$

We first note that $\mathcal{F}_{n,1}^{\max} \subset \{g \in \mathcal{N}_{n,2} : \|g\|_{\mathcal{P}} \leq 2a_{\max}\|\alpha^*\|_{\mathcal{W}(\Omega)}\}$. Since the Rademacher complexity of a set of a single function is zero, by Theorem 4.9, Proposition 4.6 and Talagrand's contraction principle (see, e.g., [49]), we have that

$$\mathcal{R}_M(\mathcal{F}_n) \leq \mathcal{R}_M(\mathcal{F}_{n,1}) \leq \sum_{i,j=1}^{N_h} \mathcal{R}_M(\mathcal{F}_{n,1}^{i,j}) \leq (N_h)^2 \mathcal{R}_M(\mathcal{F}_{n,1}^{\max}) \lesssim \frac{\kappa(A)^{1+d}}{\sqrt{M}}. \quad (4.12)$$

Therefore, from (4.6), (4.10) and (4.12), we finally obtain the following theorem, which is the main result of this paper.

Theorem 4.10. *Let Assumption 4.5 hold. If we use the $(P\ell)$ -finite element approximation, the predicted solution $u_{h,n,M}$ by the FEONet satisfies the following error estimate:*

$$\mathbb{E} [\|u - u_{h,n,M}\|_{L^1(\Omega; L^2(D))}] \lesssim h^{\ell+1} + \frac{\kappa(A)^{1+d}}{\sqrt{n}} + \frac{\kappa(A)^{1+3d/2}}{\sqrt{M}}. \quad (4.13)$$

Remark 4.11. As described earlier, in the method we propose, we first choose the finite element parameter $h > 0$ and make a triangulation. During this process, the first term in the error (4.13) can be reduced as desired, and better convergence can be achieved when using higher-order methods. After $h > 0$ is fixed and basis functions are set, we proceed with the neural network approximation for the target coefficients. Since $\kappa(A) \approx h^{-2}$, if $h > 0$ is small, the second and third terms of the error in (4.13) may increase. However, by increasing the number of neurons in the hidden layer ($n \rightarrow \infty$) and using more training samples for training ($M \rightarrow \infty$), we can reduce these error terms as much as we want. Additionally, by using the preconditioning techniques, we can significantly reduce the condition number. This provides a theory-guided strategy to reduce the second and third error terms in (4.13), ensuring faster convergence. This will be explicitly confirmed through numerical experiments in the next section.

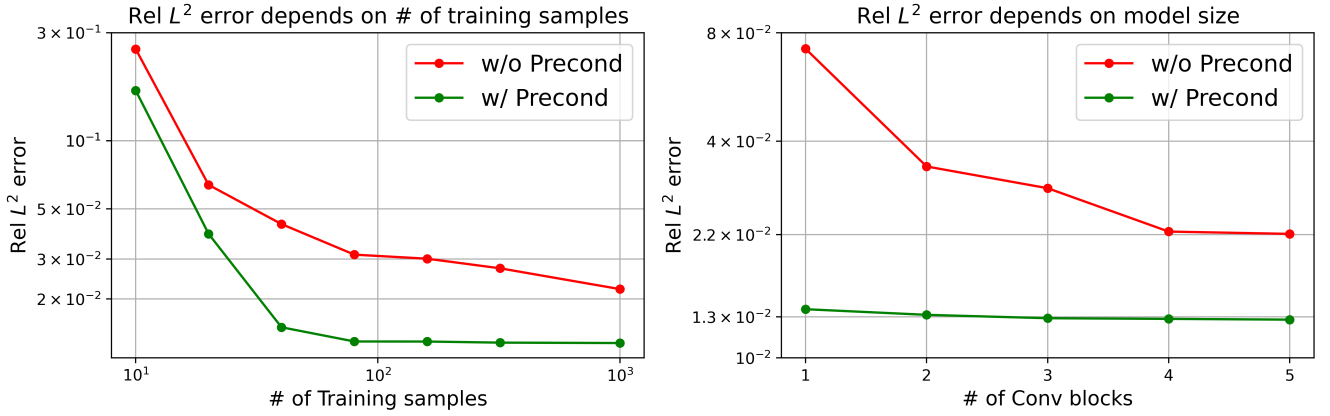


Figure 3: The relative L^2 errors resulting from varying the number of training samples and the model size.

5 Numerical experiments

In this section, we shall perform some numerical experiments to confirm the theoretical findings in the previous sections. Motivated by [3], we randomly generate input samples (in this paper, external forces) of the form

$$f(x) = n_1 \sin(m_1 \cdot x) + n_2 \cos(m_2 \cdot x) \quad x \in D, \quad (5.1)$$

where the random parameters n_1 , n_2 , m_1 and m_2 are sampled from uniform distributions. As described in Section 2.1, we train the FEONet using these random samples. Note that we don't need any precomputed input-output (f, u) pairs for training. We then evaluate the performance of the trained model using different random samples that were not used in the training. For the corresponding true solutions for the test data, we computed the finite element solutions on a sufficiently fine mesh, using the finite element software package FEniCS [35]. All the computations were conducted using the Intel Xeon Cascade Lake (Gold 6226R) processor and TESLA V100 GPU.

5.1 Convergence against the number of training samples and model size

We first demonstrate that for fixed $h > 0$, the error $\|u_h - u_{h,n,M}\|$ decreases as the model size n and the number of input samples M increase, which was theoretically proved in Theorem 3.6. To do this, we consider the 2D Poisson equation within the domain D , which is a square with a hole (see the second domain in Figure 1), i.e.,

$$\begin{aligned} -\Delta u(x, y) &= f(x, y), & (x, y) &\in D, \\ u(x, y) &= 0, & (x, y) &\in \partial D. \end{aligned} \quad (5.2)$$

Figure 3 illustrates the experimental results, where we can see a relationship between relative errors and the approximation parameters; the model size $n \in \mathbb{N}$ and the number of training samples $M \in \mathbb{N}$. We use the convolutional neural networks (CNN) as a baseline model and train the model with the gradually increasing number of convolutional blocks. We also conduct the training with a varying number of training samples from 10 to 20, 40, 80, 160, 320, and up to 1000. As we expected, we can confirm the decreasing tendency of errors as depicted with red lines in Figure 3.

Secondly, what we also proved in the theory was that the convergence depends on the condition number $\kappa(A)$ (e.g. Theorem 3.2, Theorem 3.5, (4.5) and (4.6)). This motivates us to train the FEONet

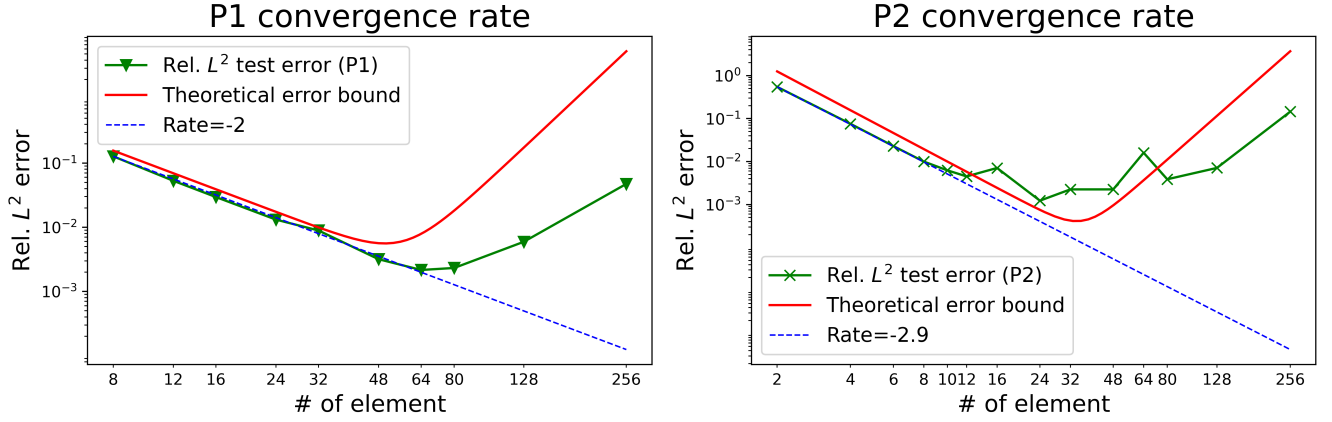


Figure 4: The relative L^2 errors against the number of elements.

with the preconditioned loss

$$\mathcal{K}(\alpha) = \|P^{-1}A\alpha(\omega) - P^{-1}F(\omega)\|_{L^1(\Omega)} \quad \text{and} \quad \mathcal{K}^M(\alpha) = \frac{|\Omega|}{M} \sum_{i=1}^M |P^{-1}A\alpha(\omega_i) - P^{-1}F(\omega_i)|, \quad (5.3)$$

so that all of $\kappa(A)$ in the theoretical results can be replaced by the term $\kappa(P^{-1}A)$ which is known to be smaller. In our experiment, for the preconditioning, we used the Sparse Approximate Inverse (SPAI) preconditioner [10]. The experimental results for this are demonstrated in Figure 3 with green lines. As we expected from theory, the errors are reduced and it has been shown that we can train the model with smaller numbers of training samples and convolutional blocks, and hence we can improve the training efficiency if we use the preconditioning. This provides us with a theory-guided classical numerical analysis strategy to improve the performance of the proposed machine-learning model.

5.2 Convergence against the number of element

Next, we shall confirm the theoretical result presented in Theorem 4.10. Note that the input samples of type (5.1) satisfy Assumption 4.5. For this experiment, we consider the convection-diffusion equation

$$\begin{aligned} -0.1u_{xx} - u_x &= f(x), & x &\in [-1, 1], \\ u(-1) &= u(1) = 0. \end{aligned} \quad (5.4)$$

The experimental results concerning the quantitative relationship between test errors and the number of elements are depicted in green in Figure 4; one for piecewise linear approximation and the other for piecewise quadratic approximation. As we can see in Figure 4, up to a certain point, as the degree of freedom increases, the relative L^2 error decreases following the convergence rate which is known in the classical theory of FEM (-2 for P1 approximation and -3 with P2 approximation), as indicated by dotted lines. But after that point, we can see that the error increases again. This phenomenon was predicted in the theoretical result presented in Theorem 3.6. Thanks to the first term in (4.13), the error goes down when it reaches a local minimum. However, as the number of elements further increases, the condition number $\kappa(A)$ goes up, and hence, so does the relative L^2 error. The theoretical error bounded for both P1 and P2 obtained in Theorem 3.6 are drawn in red in Figure 4, and we can confirm that the experimental results have a similar tendency as the theoretically predicted results.

Secondly, according to Theorem 3.6, when the number of elements is large, we can reduce the total error by using the preconditioning technique. As discussed in Section 5.1, we compare the results for the original FEONet, and the one using the preconditioned loss (5.3), and the experimental result is presented

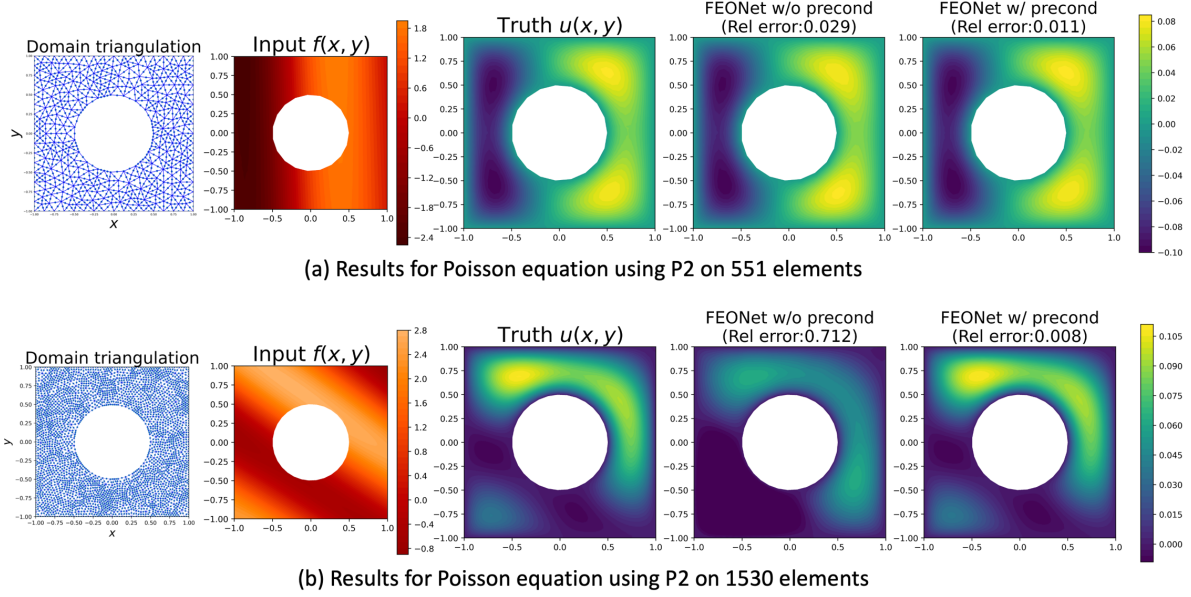


Figure 5: The FEONet prediction for 2D Poisson equation without or with preconditioning.

in Figure 5. When the number of elements is relatively small, the preconditioning slightly improves the performance. However, for the case when the number of elements is large, using the preconditioner for the training significantly improves the overall performance of the model. This also matches with the theoretical findings in the previous sections.

6 Conclusion

In this paper, we have investigated the convergence of approximate solutions predicted by the FEONet proposed in [32], and derived an error estimate, simultaneously examining the whole approximation parameters $h > 0$ and $n, M \in \mathbb{N}$. Based on the eigenvalue estimates from the classical FEM theory, we proved the convergence of the numerical solution for general linear second-order PDEs, and the convergence depends on the condition number of the finite element matrix. Furthermore, for a self-adjoint case, a novel regularity theory for the neural network approximation was proposed, from which we obtained the complete error estimate. Finally, we conducted some numerical experiments which support the theoretically shown results in the paper.

An interesting future research direction is to study the optimization error. Since the neural tangent kernel (see e.g., [23]) of the FEONet also depends on the finite element matrices, again from the eigenvalue estimates for FEM, we may theoretically analyze the corresponding optimization error. The analysis of the FEONet for nonlinear equations is also intriguing. In [32], we showed that the FEONet can also learn a solution operator for nonlinear equations. This ability becomes particularly highlighted if we compare the FEONet with the classical FEM, since we don't need any iterative schemes to predict solutions, allowing us to make a real-time solution prediction for nonlinear equations. These topics are of independent interest, and will be addressed in the forthcoming papers.

References

- [1] M. Ainsworth, B. McLean, and T. Tran. Diagonal scaling of stiffness matrices in the Galerkin boundary element method. *ANZIAM J.*, 42(1):141–150, 2000. Papers in honour of David Elliott on

the occasion of his sixty-fifth birthday.

- [2] M. Ainsworth, W. McLean, and T. Tran. The conditioning of boundary element equations on locally refined meshes and preconditioning by diagonal scaling. *SIAM J. Numer. Anal.*, 36(6):1901–1932, 1999.
- [3] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.
- [4] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.
- [5] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, Jan 1994.
- [6] N. Boullé and A. Townsend. A mathematical guide to operator learning. *arXiv preprint arXiv:2312.14688*, 2023.
- [7] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.
- [8] Z. Chen, J. Lu, and Y. Lu. On the representation of solutions to elliptic pdes in barron spaces. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 6454–6465, 2021.
- [9] J. Choi, T. Yun, N. Kim, and Y. Hong. Spectral operator learning for parametric pdes without data reliance. *Computer Methods in Applied Mechanics and Engineering*, 420:116678, 2024.
- [10] E. Chow and Y. Saad. Approximate inverse preconditioners via sparse-sparse iterations. *SIAM Journal on Scientific Computing*, 19(3):995–1023, 1998.
- [11] F. S. Costabal, S. Pezzuto, and P. Perdikaris. δ -pinns: Physics-informed neural networks on complex geometries. *Engineering Applications of Artificial Intelligence*, 127:107324, 2024.
- [12] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, 2(4):303–314, 1989.
- [13] T. De Ryck, S. Mishra, and R. Molinaro. wpinns: Weak physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 62(2):811–841, 2024.
- [14] W. E, C. Ma, and L. Wu. The Barron space and the flow-induced function spaces for neural network models. *Constr. Approx.*, 55(1):369–406, 2022.
- [15] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. Dnn expression rate analysis of high-dimensional pdes: Application to option pricing. *Constructive Approximation*, 55(1):3–71, 2022.
- [16] A. Ern and J.-L. Guermond. Evaluation of the condition number in linear systems arising in finite element approximations. *M2AN Math. Model. Numer. Anal.*, 40(1):29–48, 2006.
- [17] G. Gnecco and M. Sanguineti. Approximation error bounds via Rademacher’s complexity. *Appl. Math. Sci. (Ruse)*, 2(1-4):153–176, 2008.
- [18] I. G. Graham and W. McLean. Anisotropic mesh refinement: the conditioning of Galerkin boundary element matrices and simple preconditioners. *SIAM J. Numer. Anal.*, 44(4):1487–1513, 2006.

- [19] Y. Gu and M. K. Ng. Deep adaptive basis galerkin method for high-dimensional evolution equations with oscillatory solutions. *SIAM Journal on Scientific Computing*, 44(5):A3130–A3157, 2022.
- [20] Q. Hong, J. W. Siegel, and J. Xu. A priori analysis of stable neural network solutions to numerical pdes, 2021.
- [21] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, second edition, 2013.
- [22] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [23] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [24] A. D. Jagtap and G. E. Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.
- [25] L. Kamenski. Sharp bounds on the smallest eigenvalue of finite element equations with arbitrary meshes without regularity assumptions. *SIAM J. Numer. Anal.*, 59(2):983–997, 2021.
- [26] P. Kidger and T. Lyons. Universal Approximation with Deep Narrow Networks. In J. Abernethy and S. Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2306–2327. PMLR, 09–12 Jul 2020.
- [27] S. Ko, S.-B. Yun, and Y. Hong. Convergence analysis of unsupervised legendre-galerkin neural networks for linear second-order elliptic pdes. *arXiv preprint arXiv:2211.08900*, 2022.
- [28] N. Kovachki, S. Lanthaler, and S. Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.
- [29] S. Lanthaler, S. Mishra, and G. E. Karniadakis. Error estimates for deepoanets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1), 2022.
- [30] J. Y. Lee, S. CHO, and H. J. Hwang. HyperdeepONet: learning operator with complex target function space using the limited resources via hypernetwork. In *The Eleventh International Conference on Learning Representations*, 2023.
- [31] J. Y. Lee, J. Jang, and H. J. Hwang. opPINN: physics-informed neural network with operator learning to approximate solutions to the Fokker-Planck-Landau equation. *J. Comput. Phys.*, 480:Paper No. 112031, 21, 2023.
- [32] J. Y. Lee, S. Ko, and Y. Hong. Finite element operator network for solving parametric pdes. *arXiv preprint arXiv:2308.04690*, 2023.
- [33] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.
- [34] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 2021.

- [35] A. Logg and G. N. Wells. Dofin: Automated finite element computing. *ACM Trans. Math. Softw.*, 37(2), apr 2010.
- [36] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.
- [37] 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 M. Belkin and S. Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3196–3241. PMLR, 15–19 Aug 2021.
- [38] C. Marcati and C. Schwab. Exponential convergence of deep operator networks for elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 61(3):1513–1545, 2023.
- [39] T. Marwah, Z. C. Lipton, J. Lu, and A. Risteski. Neural network approximations of pdes beyond linearity: A representational perspective. In *International Conference on Machine Learning*, pages 24139–24172. PMLR, 2023.
- [40] S. Mishra and R. Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating pdes. *IMA Journal of Numerical Analysis*, 43(1):1–43, 2023.
- [41] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *CoRR*, abs/1805.12076, 2018.
- [42] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In P. Grünwald, E. Hazan, and S. Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1376–1401, Paris, France, 03–06 Jul 2015. PMLR.
- [43] G. Ongie, R. Willett, D. Soudry, and N. Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International Conference on Learning Representations*, 2020.
- [44] R. Parhi and R. D. Nowak. Banach space representer theorems for neural networks and ridge splines. *Journal of Machine Learning Research*, 22(43):1–40, 2021.
- [45] 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. *Journal of Computational physics*, 378:686–707, 2019.
- [46] 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(5):2042–2074, 2020.
- [47] J. W. Siegel and J. Xu. Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321, 2020.
- [48] J. W. Siegel and J. Xu. High-order approximation rates for shallow neural networks with cosine and reluk activation functions. *Applied and Computational Harmonic Analysis*, 58:1–26, 2022.
- [49] M. J. Wainwright. *High-dimensional statistics*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2019. A non-asymptotic viewpoint.

- [50] S. Wang, H. Wang, and P. Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deepnets. *Science advances*, 7(40), 2021.
- [51] J. Xu. Finite neuron method and convergence analysis. *Commun. Comput. Phys.*, 28(5):1707–1745, 2020.
- [52] L. Yang, X. Meng, and G. E. Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.