

DEEP RITZ METHOD FOR THE SPECTRAL FRACTIONAL LAPLACIAN EQUATION USING THE CAFFARELLI-SILVESTRE EXTENSION

YIQI GU* AND MICHEAL K. NG*

Abstract. In this paper, we propose a novel method for solving high-dimensional spectral fractional Laplacian equations. Using the Caffarelli-Silvestre extension, the d -dimensional spectral fractional equation is reformulated as a regular partial differential equation of dimension $d + 1$. We transform the extended equation as a minimal Ritz energy functional problem and search for its minimizer in a special class of deep neural networks. Moreover, based on the approximation property of networks, we establish estimates on the error made by the deep Ritz method. Numerical results are reported to demonstrate the effectiveness of the proposed method for solving fractional Laplacian equations up to ten dimensions. Technically, in this method, we design a special network-based structure to adapt to the singularity and exponential decaying of the true solution. Also, A hybrid integration technique combining Monte Carlo method and sinc quadrature is developed to compute the loss function with higher accuracy.

Key words. Ritz method; Deep learning; Fractional Laplacian; Caffarelli-Silvestre extension; Singularity;

AMS subject classifications. 65N15; 65N30; 68T07; 41A25

1. Introduction. As a nonlocal generalization of the Laplacian $-\Delta$, the spectral fractional Laplacian $(-\Delta)^s$ with a fraction s arises in many areas of applications, such as anomalous diffusion [10, 30], turbulent flows [26], Lévy processes [16], quantum mechanics [18], finance [29, 12] and pollutant transport [31]. In this paper, we develop a network-based Ritz method for solving fractional Laplacian equations using the Caffarelli-Silvestre extension.

Let $d \in \mathbb{N}^+$ be the dimension of the problem and Ω be a bounded Lipschitz domain in \mathbb{R}^d . Also, suppose $0 < s < 1$ and f is a function defined in Ω , we consider the following spectral fractional Laplacian equation with homogeneous Dirichlet condition,

$$(1.1) \quad \begin{cases} (-\Delta)^s U(x) = f(x), & \forall x \in \Omega, \\ U(x) = 0, & \forall x \in \partial\Omega, \end{cases}$$

where $(-\Delta)^s$ is defined by the spectral decomposition of $-\Delta$ with the same boundary conditions. More precisely, we suppose the countable set $\{(\lambda_n, \phi_n)\}_{n=1}^\infty$ are all the eigenvalues and orthonormal eigenfunctions of the following problem,

$$(1.2) \quad \begin{cases} -\Delta\phi = \lambda\phi, & \text{in } \Omega, \\ \phi = 0, & \text{on } \partial\Omega, \\ (\phi, \phi) = 1, \end{cases}$$

where (\cdot, \cdot) is the standard inner product in $L^2(\Omega)$. Then for any $U \in L^2(\Omega)$,

$$(1.3) \quad (-\Delta)^s U = \sum_{n=1}^{\infty} \tilde{U}_n \lambda_n^s \phi_n, \text{ with } \tilde{U}_n := (U, \phi_n).$$

We remark that another definition of the fractional Laplacian is formulated by integrals with non-local structures, and these two definitions do not coincide. It is difficult to solve fractional

*Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong (yiqigu@hku.hk, mng@maths.hku.hk). This work is supported by HKRGC GRF 12300218, 12300519, 17201020 and 17300021.

Laplacian equations of either definition directly using numerical methods for regular differential equations (e.g., finite difference method or finite element method) due to the non-local property of the fractional operator [5, 1]. Instead, one effective approach is to use the Caffarelli-Silvestre extension [7]. Specifically, let us introduce a scalar variable y and consider the following $d + 1$ -dimensional problem

$$(1.4) \quad \begin{cases} \nabla \cdot (y^\alpha \nabla u(x, y)) = 0, & \forall (x, y) \in \mathcal{D} := \Omega \times (0, \infty), \\ -\lim_{y \rightarrow 0} y^\alpha u_y(x, y) = d_s f(x), & \forall x \in \Omega, \\ u(x, y) = 0, & \forall (x, y) \in \partial_L \mathcal{D} := \partial\Omega \times [0, \infty), \end{cases}$$

where $\alpha = 1 - 2s$ and $d_s = 2^{1-2s}\Gamma(1-s)/\Gamma(s)$. Suppose $u(x, y)$ solves (1.4), then $U(x) := u(x, 0)$ is a solution of (1.1) [24]. Consequently, one can solve the extended problem (1.4) with regular derivatives to avoid addressing spectral fractional differential operators, with the extra cost that (i) the dimension is increased from d to $d + 1$; (ii) the domain is extended from the bounded one Ω to an unbounded cylinder $\partial_L \mathcal{D}$. Several methods have been proposed for (1.4), such as the tensor product finite elements [24] and the enriched spectral method using Laguerre functions [11]. We remark that the Caffarelli-Silvestre extension is exclusively for the spectral fractional Laplacian and not for the integral fractional Laplacian. Moreover, the extension technique can be extended to more general fractional symmetric elliptic operators of the form $-\nabla \cdot (A(x)\nabla U(x)) + C(x)U(x)$ with $A(x)$ being symmetric and positive definite and $C(x)$ being positive.

However, conventional linear structures such as finite elements and polynomials are usually incapable of high-dimensional approximation in practice. For example, suppose a tensor product linear structure has \tilde{N} basis functions in each dimension, then the total degree of freedom is $O(\tilde{N}^d)$, which is a huge number if d is moderately large. Such a curse of dimensionality prevents one from using linear algebra structures in high-dimensional problems with $d > 3$. For the spectral fractional Laplacian, most existing methods based on the Caffarelli-Silvestre extension could solve numerical examples of dimension at most two, mainly due to the limitation of storage. Our primary target is to solve many physically relevant problems that appear in three or higher dimensional situations.

In recent years, deep neural networks (DNNs) are widely studied and utilized in applied problems. As a composition of simple neural functions, the DNN has parameters nonlinearly arrayed in the network structure. For a fully-connected DNN with depth L , width M and dimension of inputs d , the total number of parameters is of $O(LM^2 + Md)$. Therefore, the degree of freedom increases linearly with the dimension and DNNs are capable of dealing with high-dimensional approximation in practice. Theoretically, it is shown that DNNs have decent approximation properties for particular function spaces (e.g., Barron space). The seminal work of Barron [2, 3] deduces L^2 -norm and L^∞ -norm approximations of two-layer sigmoid networks. Recent work [17, 14, 15, 13, 27, 28, 9, 19] considers more variants of the network-based approximation for Barron-type spaces. Generally, given a Barron function $g : \Omega \rightarrow \mathbb{R}$, there exists a two-layer neural network g_M with width M and common-used activations such that

$$(1.5) \quad \|g - g_M\|_\Omega \leq O\left(\frac{\|g\|_{\mathcal{B}}}{\sqrt{M}}\right),$$

where $\|g\|_{\mathcal{B}}$ is the Barron norm of g , and $\|\cdot\|_\Omega$ can be $L^2(\Omega)$, $H^1(\Omega)$ or $L^\infty(\Omega)$ norm under different hypothesis. It is worth mentioning that the above error bound is independent of the

dimension of the input variable; hence the network-based approximation can overcome the curse of dimensionality.

In this paper, we solve (1.4) by Ritz method in which DNNs are employed to approximate the solution. More precisely, we reformulate (1.4) as a minimal Ritz energy functional problem and characterize the Sobolev space of the weak solution. Next, as a subset of the solution space, a class of DNNs is taken as the hypothesis space of the minimization. We design a special network structure for the DNN class such that (i) it satisfies the homogeneous boundary condition on $\partial_L \mathcal{D}$ in (1.4); (ii) it decays to zero exponentially as $y \rightarrow \infty$; and (iii) it has a singularity behaves as $O(y^{k+1-\alpha})$ for integers k at $y = 0$. Note that the second and third properties mentioned above are also satisfied by the true solution. Consequently, the special DNNs have better approximation performance than generic DNNs, which is also observed in our numerical experiments.

Theoretically, under a Barron-type framework, we investigate the approximation error between the special DNN class and the solution space under the Sobolev norm, which has a similar form to (1.5). Based on that, we estimate the solution error of the proposed Ritz method using the special DNN class, assuming that the true solution has components in the Barron space. The final solution error is of $O(M^{-1/2})$, which is consistent with the approximation error. We remark that the error bound of our method is advantageous over the existing methods [24, 11] using finite element or Laguerre functions if the dimension is moderately large, since the error order $-1/2$ is independent of the dimension. Also, the error order is consistent with the deep Ritz method for regular Laplacian equations [23]

In the implementation, a combination of stochastic and deterministic methods is employed to compute the integrals in the energy functional. Specifically, we utilize the quasi-Monte Carlo method and the sinc quadrature rule [21] to evaluate the integrals in terms of x and y , respectively. For the former, due to the potentially high dimension of x , Monte Carlo-type methods are effective and easy to implement. For the latter, although the integrand in terms of y is one-dimensional, it has a singular term y^α when $\alpha \neq 0$. While sinc quadrature is highly accurate for integrals with fractional powers and therefore preferred here. By numerical experiments, we demonstrate that our method can solve model problems up to $d = 10$ with desired accuracy. To the best of our knowledge, this is the first attempt to solve high-dimensional fractional Laplacian equations by deep learning methods.

Overall, the highlights of our work can be summarized as follows:

- Development of a special approximation structure based on generic DNNs according to the special properties of the true solution;
- Combination of stochastic Monte Carlo method for high dimensions and deterministic sinc quadrature for high accuracy in the learning process;
- Simulation of 10-D fractional Laplacian equations with relative error $O(10^{-2})$.

The rest of the paper is organized as follows. In Section 2, we reformulate the problem (1.4) as the minimization of an energy functional and show their equivalence. In Section 3, the fully connected neural networks are introduced. We characterize the special structures of the hypothesis space and discuss its approximation property. In Section 4, we derive the error estimate for the proposed method. Numerical experiments are presented to show the effectiveness of our method in Section 5. Finally, some concluding remarks are given in Section 6.

2. Minimization of Energy Functional. We solve the regular partial differential equation (1.4) under the framework of Ritz method. The equation can be transformed to an equivalent minimal functional, and we look for Sobolev weak solutions instead of classical solutions. Simi-

larly, one can also solve (1.4) using Galerkin method by introducing appropriate test spaces such as in [24, 11]. Since learning-based methods aim to find solutions via optimization, the use of Ritz method can be achieved for building such formulation.

2.1. The space of weak solutions. Let \mathcal{Z} be any region and ω be a positive weight function. We define the weighted L^2 space as

$$(2.1) \quad L_\omega^2(\mathcal{Z}) := \left\{ v \mid \int_{\mathcal{Z}} |v(z)|^2 \omega(z) dz < \infty \right\},$$

equipped with the inner product

$$(2.2) \quad (v_1, v_2)_{\omega, \mathcal{Z}} := \int_{\mathcal{Z}} v_1(z) v_2(z) \omega(z) dz, \quad \forall v_1, v_2 \in L_\omega^2(\mathcal{Z}),$$

and the induced norm

$$(2.3) \quad \|v\|_{\omega, \mathcal{Z}} := (v, v)_{\omega, \mathcal{Z}}^{\frac{1}{2}}.$$

The weight ω is omitted from the notations if $\omega \equiv 1$.

We also define the weighted Sobolev space as

$$(2.4) \quad H_\omega^1(\mathcal{Z}) := \{v \mid v \in L_\omega^2(\mathcal{Z}), \nabla v \in L_\omega^2(\mathcal{Z})\},$$

equipped with the norm

$$(2.5) \quad \|v\|_{1, \omega, \mathcal{Z}} := \left(\|v\|_{\omega, \mathcal{Z}}^2 + \|\nabla v\|_{\omega, \mathcal{Z}}^2 \right)^{\frac{1}{2}}.$$

It is shown in [22] the solution of the extended problem (1.4) has a desirable property that it converges exponentially to zero as $y \rightarrow \infty$. Therefore we can define the solution space as

$$(2.6) \quad H_{y^\alpha}^{1, \text{b}}(\mathcal{D}) := \left\{ v \in H_{y^\alpha}^1(\mathcal{D}) \mid \lim_{y \rightarrow \infty} v(x, y) = 0, v(x, y)|_{\partial_L \mathcal{D}} = 0 \right\},$$

with the norm

$$(2.7) \quad \|v\|_{H_{y^\alpha}^{1, \text{b}}(\mathcal{D})} = \|\nabla v\|_{y^\alpha, \mathcal{D}}.$$

Denote the trace for all $v \in H_{y^\alpha}^{1, \text{b}}(\mathcal{D})$ by

$$(2.8) \quad \text{tr}\{v\}(x) = v(x, 0).$$

Moreover, for column vectors or vector-valued functions, we use $|\cdot|$ to denote their Euclidean norm.

2.2. Minimal energy functional. We aim to characterize the solution of (1.4) as a minimizer of an corresponding energy functional. For this, we define the functional

$$(2.9) \quad \mathcal{I}[w] := \frac{1}{2}(\nabla w, \nabla w)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{w\})_\Omega, \quad \forall w \in H_{y^\alpha}^{1,b}(\mathcal{D}).$$

We have the following result.

THEOREM 2.1. *Assume $u \in H_{y^\alpha}^{1,b}(\mathcal{D})$ solves (1.4). Then*

$$(2.10) \quad \mathcal{I}[u] = \min_{w \in H_{y^\alpha}^{1,b}(\mathcal{D})} \mathcal{I}[w].$$

Conversely, if u satisfies (2.10), then u solves the problem (1.4).

Proof. To prove (2.10), for all $w \in H_{y^\alpha}^{1,b}(\mathcal{D})$, $u - w \in H_{y^\alpha}^{1,b}(\mathcal{D})$. Then using the fact that $(u - w)|_{\partial_L \mathcal{D}} = (u - w)|_{\Omega \times \{+\infty\}} = 0$ and integration by parts we have

$$(2.11) \quad (\nabla \cdot (y^\alpha \nabla u), u - w)_\mathcal{D} = -(y^\alpha \nabla u, \nabla(u - w))_\mathcal{D} + (-y^\alpha u_y, u - w)_{\Omega \times \{0\}}.$$

Note the left hand side is equal to zero since $\nabla \cdot (y^\alpha \nabla u) \equiv 0$. And the second term on the left is

$$(2.12) \quad \begin{aligned} (-y^\alpha u_y, u - w)_{\Omega \times \{0\}} &= (-y^\alpha u_y|_{y=0}, u(x, 0) - w(x, 0))_\Omega \\ &= \left(\lim_{y \rightarrow 0} (-y^\alpha u_y), u(x, 0) - w(x, 0) \right)_\Omega = (d_s f(x), \text{tr}\{u\} - \text{tr}\{w\})_\Omega. \end{aligned}$$

Therefore, it follows (2.11)

$$(2.13) \quad (y^\alpha \nabla u, \nabla(u - w))_\mathcal{D} = (d_s f(x), \text{tr}\{u\} - \text{tr}\{w\})_\Omega,$$

which implies

$$(2.14) \quad (\nabla u, \nabla u)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{u\})_\Omega = (\nabla u, \nabla w)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{w\})_\Omega.$$

Using the inequality $\nabla u \cdot \nabla w \leq \frac{1}{2}|\nabla u|^2 + \frac{1}{2}|\nabla w|^2$, it leads to

$$(2.15) \quad \mathcal{I}[u] \leq \mathcal{I}[w].$$

On the other hand, suppose (2.10) holds. Fix any $w \in H_{y^\alpha}^{1,b}(\mathcal{D})$ and write

$$(2.16) \quad i(\tau) := \mathcal{I}[u + \tau w], \quad \forall \tau \in \mathbb{R}.$$

Note

$$(2.17) \quad \begin{aligned} i(\tau) &= \frac{1}{2}(\nabla(u + \tau w), \nabla(u + \tau w))_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{u + \tau w\})_\Omega \\ &= \frac{1}{2}(\nabla u, \nabla u)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{u\})_\Omega + \tau \left((\nabla u, \nabla w)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{w\})_\Omega \right) + \frac{1}{2}\tau^2 (\nabla w, \nabla w)_{y^\alpha, \mathcal{D}}. \end{aligned}$$

Since $u + \tau w \in H_{y^\alpha}^{1,b}(\mathcal{D})$ for each τ , $i(\tau)$ takes its minimum at $\tau = 0$, and thus

$$(2.18) \quad 0 = i'(0) = (\nabla u, \nabla w)_{y^\alpha, \mathcal{D}} - d_s(f, \text{tr}\{w\})_\Omega.$$

Using integration by parts we have

$$(2.19) \quad (\nabla \cdot (y^\alpha \nabla u), w)_{\mathcal{D}} + \left(d_s f + \lim_{y \rightarrow 0} y^\alpha u_y, \text{tr}\{w\} \right)_{\Omega} = 0.$$

Especially, (2.19) holds for all $w \in C_c^\infty(\mathcal{D})$, which implies

$$(2.20) \quad \int_{\mathcal{D}} (\nabla \cdot (y^\alpha \nabla u)) w = 0, \quad \forall w \in C_c^\infty(\mathcal{D}),$$

leading to $\nabla \cdot (y^\alpha \nabla u) = 0$ in \mathcal{D} . And thus by (2.19)

$$(2.21) \quad \left(d_s f + \lim_{y \rightarrow 0} y^\alpha u_y, \text{tr}\{w\} \right)_{\Omega} = 0.$$

Especially, $\text{tr}\{w\}$ takes over all functions in $C_c^\infty(\Omega)$, which leads to $d_s f + \lim_{y \rightarrow 0} y^\alpha u_y = 0$ in Ω . \square

By virtue of Theorem 2.1, it suffices to solve the following optimization

$$(2.22) \quad \min_{w \in H_{y^\alpha}^{1,b}(\mathcal{D})} \mathcal{I}[w],$$

whose solution is exactly a weak solution of the extended problem (1.4).

3. Neural Network Approximation. In the numerical computation, one aims to introduce a function set with a finite degree of freedom to approximate the solution space $H_{y^\alpha}^{1,b}(\mathcal{D})$, and minimize $\mathcal{I}[\cdot]$ in this appropriate set of functions. In many physical relevant problems, it is required to address $d \geq 3$, causing the dimension of \mathcal{D} no less than 4. Potentially high dimensions impede the usage of conventional linear structures. However, as a nonlinear structure, DNNs can approximate high-dimensional functions by moderately less degree of freedom. This inspires us to use classes of DNNs to approximate $H_{y^\alpha}^{1,b}(\mathcal{D})$ especially when d is large.

3.1. Fully connected neural network. In our method, we employ the fully connected neural network (FNN) which is one of the most common neural networks in deep learning. Mathematically speaking, let $\sigma(t)$ be an activation function which is applied entry-wise to a vector x to obtain another vector of the same size. Let $L \in \mathbb{N}^+$ and $M_\ell \in \mathbb{N}^+$ for $\ell = 1, \dots, L$, an FNN $\hat{\phi}$ is the composition of L simple nonlinear functions as follows

$$(3.1) \quad \hat{\phi}(z; \theta) := a^T h_L \circ h_{L-1} \circ \dots \circ h_1(z) \quad \text{for } z \in \mathbb{R}^d,$$

where $a \in \mathbb{R}^{M_L}$; $h_\ell(z_\ell) := \sigma(W_\ell z_\ell + b_\ell)$ with $W_\ell \in \mathbb{R}^{M_\ell \times M_{\ell-1}}$ ($M_0 := d$) and $b_\ell \in \mathbb{R}^{M_\ell}$ for $\ell = 1, \dots, L$. Here M_ℓ is called the width of the ℓ -th layer and L is called the depth of the FNN. $\theta := \{a, W_\ell, b_\ell : 1 \leq \ell \leq L\}$ is the set of all parameters in $\hat{\phi}$ to determine the underlying neural network. Common types of activation functions include the sigmoid function $(1 + e^{-t})^{-1}$ and the rectified linear unit (ReLU) $\max(0, t)$. We remark that, when solving k -th order differential equations, many existing network-based methods use the ReLU^{k+1} activation function $\max\{0, t^{k+1}\}$, so that their networks are C^k functions and can be applied by the differential operators. While in the minimization (2.22), only H^1 regularity is required and therefore ReLU networks suffice.

Denote $M := \max \{M_\ell, 1 \leq \ell \leq L\}$, then it is clear that $|\theta| = O(M^2 L + M d)$. Comparatively, the degree of freedom of linear structures such as finite elements and tensor product polynomials increases exponentially with d . Hence FNNs are more practicable in high-dimensional approximations. For simplicity, we consider the architecture $M = M_\ell$ for all ℓ and denote $\mathcal{F}_{L,M,\sigma}$ as the set consisting of all FNNs with depth L , width M and activation function σ . In the following passage, all functions involving an FNN will be denoted with the superscript $\hat{\cdot}$.

3.2. Special structures of the approximate class. Recent work [20, 25] indicates that deep FNNs can approximate smooth functions in ∞ -norm within any desired accuracy as long as the depth or width is large enough. The approximation will be more accurate if the target function has higher regularity. However, it is shown in [8, 11] that the solution of (1.4) has a singularity at $y = 0$ which behaves as $y^{k+1-\alpha}$ for $k \in \mathbb{N}^+$. Therefore, it is not appropriate to naively use the class of generic FNNs. Instead, We aim to develop a special structure based on FNNs for the approximate class.

In the enriched spectral method [11], the solution of (2.19) is approximated by a structure consisting of two parts. One part is a linear combination of smooth basis functions, which approximates the regular component of the solution. The other part is a linear tensor product combination of smooth basis functions and a sequence of artificial terms $\{y^{k-\alpha}\}_{k=1,2,\dots}$, which is for the singular component of the solution. Following this idea, we use $\hat{\phi}(x, y)$ to denote any function in the approximate class, and build its structure as the combination of two parts,

$$(3.2) \quad \hat{\phi}(x, y) = \hat{\phi}_1(x, y) + y^{1-\alpha} \hat{\phi}_2(x, y),$$

where $\hat{\phi}_1, \hat{\phi}_2$ are FNN-based smooth functions and the term $y^{1-\alpha}$ is introduced to adapt to the singularity at $y = 0$.

Moreover, since the true solution of the extended problem (1.4) converges to zero as $y \rightarrow \infty$, the functions in the approximate class should also preserve this property. To realize it, we can introduce exponential terms concerning y in the structure of $\hat{\phi}(x, y)$. Specifically, we let

$$(3.3) \quad \hat{\phi}_1(x, y) = \hat{\phi}_3(x, y)e^{-\gamma'y}, \quad \hat{\phi}_2(x, y) = \hat{\phi}_4(x, y)e^{-\gamma''y},$$

where $\{\hat{\phi}_3, \hat{\phi}_4\}$ are FNN-based functions and $\{\gamma', \gamma''\} > 0$ are two auxiliary scalar parameters ensuring that $\hat{\phi}_1$ and $\hat{\phi}_2$ converges to zero as $y \rightarrow \infty$ exponentially.

In the end, as a subset of $H_{y^\alpha}^{1,b}(\mathcal{D})$, the approximate class should also consist of functions satisfying the boundary condition; namely, $\hat{\phi}|_{\partial_L \mathcal{D}} = 0$. We achieve this by setting

$$(3.4) \quad \hat{\phi}_3(x, y) = \hat{\phi}'(x, y)h(x), \quad \hat{\phi}_4(x, y) = \hat{\phi}''(x, y)h(x),$$

where $\{\hat{\phi}', \hat{\phi}''\}$ are generic FNNs and $h(x)$ is a smooth function constructed particularly to satisfy $h(x) = 0$ on $\partial\Omega$. For example, if Ω is a hypercube $(a_1, b_1) \times \dots \times (a_d, b_d)$, then $h(x)$ can be chosen as $h(x) = \prod_{i=1}^d (x_i - a_i)(x_i - b_i)$; if Ω has a boundary characterized by a level set, say $\partial\Omega = \{x \mid \rho(x) = 0\}$, for some continuous function ρ , then $h(x)$ can be chosen as $h(x) = \rho(x)$. Generally, we can set $h(x) = F(\text{dist}(x, \partial\Omega))$, where F is a particular analytic function satisfying $F(0) = 0$ and $\text{dist}(x, \partial\Omega)$ represents the distant between x and $\partial\Omega$. In practice, we expect to set $h(x)$ as smooth as possible so that $\hat{\phi}_3$ and $\hat{\phi}_4$ preserve the same regularity as $\hat{\phi}'$ and $\hat{\phi}''$, respectively.

To sum up, we build the following special structure for the approximate class,

$$(3.5) \quad \hat{\phi}((x, y); \theta) = \hat{\phi}'((x, y); \theta')h(x)e^{-\gamma'y} + y^{1-\alpha} \hat{\phi}''((x, y); \theta'')h(x)e^{-\gamma''y},$$

where $\theta := \{\theta', \theta'', \gamma', \gamma''\}$ is the set of all trainable parameters. We denote $\mathcal{N}_{L,M,\sigma,h}$ as the class of all neural networks having the structure in (3.5), namely,

$$(3.6) \quad \mathcal{N}_{L,M,\sigma,h} = \left\{ \hat{\phi} : \mathcal{D} \rightarrow \mathbb{R} \mid \hat{\phi}(x, y) = \hat{\phi}'(x, y)h(x)e^{-\gamma'y} + y^{1-\alpha} \hat{\phi}''(x, y)h(x)e^{-\gamma''y}, \right. \\ \left. \forall \hat{\phi}', \hat{\phi}'' \in \mathcal{F}_{L,M,\sigma}, \forall \gamma', \gamma'' \in \mathbb{R}^+ \right\}.$$

It is clear that $\mathcal{N}_{L,M,\sigma,h}$ is a subset of $H_{y^\alpha}^{1,b}(\mathcal{D})$ as long as σ is Lipschitz continuous and has a polynomial growth bound; namely, $|\sigma(t)| \leq C(1 + |t|^p)$ for all $t \in \mathbb{R}$ with a constant $C > 0$ and an integer $p > 0$. In our Ritz method, $\mathcal{N}_{L,M,\sigma,h}$ is taken as the approximate class of $H_{y^\alpha}^{1,b}(\mathcal{D})$.

3.3. Approximation property. We will show that functions in $H_{y^\alpha}^{1,b}(\mathcal{D})$ can be approximated by the special networks in $\mathcal{N}_{L,M,\sigma,h}$ as $M \rightarrow \infty$. To illustrate the approximation property, we first introduce the Barron space and then derive the error bounds for neural-network approximation, assuming that the target function has components in the Barron space. In this section, we specifically focus on the FNNs with ReLU activation and specify $\sigma(x) = \max(x, 0)$. For simplicity, we concatenate variables $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$ by writing $z = (x, y) \in \mathbb{R}^{d+1}$.

3.3.1. Barron space. Let us first quickly review the Barron space and norm. We will focus on the definition discussed in [15] which represents infinitely wide two-layer ReLU FNNs. Following Section 3.1, recall the set of two-layer ReLU FNNs without output bias is given by

$$(3.7) \quad \mathcal{F}_{2,M,\text{ReLU}} = \left\{ \hat{\phi} \mid \hat{\phi}(z) = \frac{1}{M} \sum_{i=1}^M a_i \sigma(b_i^T z + c_i), \quad \forall (a_i, b_i, c_i) \in \mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{R} \right\}.$$

For a probability measure π on $\mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{R}$, we set the function

$$(3.8) \quad f_\pi(z) = \int_{\mathcal{D}} a \sigma(b^T z + c) \pi(da, db, dc) = \mathbb{E}_\pi[a \sigma(b^T z + c)], \quad \forall z \in \mathbb{R}^{d+1},$$

given this expression exists. For a function $u : \mathbb{R}^{d+1} \rightarrow \mathbb{R}$, we use Π_u to denote the set of all probability measures π such that $f_\pi(z) = u(z)$ almost everywhere. Then the Barron norm is defined as

$$(3.9) \quad \|u\|_{\mathcal{B}}^2 := \inf_{\pi \in \Pi_u} \int_{\mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{R}} a^2(|b| + |c|)^2 \pi(da, db, dc) = \inf_{\pi \in \Pi_u} \mathbb{E}_\pi[a^2(|b| + |c|)^2].$$

The infimum of the empty set is considered as $+\infty$. The set of all functions with finite Barron norm is denoted by \mathcal{B} . It is shown in [15] that \mathcal{B} equipped with the Barron norm is a Banach space which is called Barron space.

3.4. Error estimation. Let u be a function in $H_{y^\alpha}^{1,b}(\mathcal{D})$, and further assume that u converges to zero exponentially as $y \rightarrow \infty$. In the error analysis, we make the following assumption that u can be factorized explicitly by components vanishing on $\partial_L \mathcal{D}$ and decaying to zero exponentially as $y \rightarrow \infty$.

ASSUMPTION 3.1. *There exist functions $h \in H_0^1(\Omega)$, $v \in \mathcal{B}$ and some number $\eta > 0$ such that*

$$(3.10) \quad u(z) = v(z)h(x)e^{-\eta y}.$$

Especially, if Assumption 3.1 holds, we can normalize h such that $|h| \leq 1$ and $|\nabla h| \leq 1$. Assumption 3.1 is indeed satisfied in some situations. For example, in the case $\Omega = [-1, 1]^2$, $s = 0.5$ and $f = \sqrt{2}\pi \sin(\pi x_1) \sin(\pi x_2)$, the solution u of (1.4) is given by $u = \sin(\pi x_1) \sin(\pi x_2) e^{-\sqrt{2}\pi y}$. Note that by Taylor series $\sin(\pi x) = -\frac{\pi}{2}(x^2 - 1) + \frac{\pi}{8}(x^2 - 1)^2 + O((x^2 - 1)^3)$, u can be written as (3.10) with

$$v = \left(-\pi + \frac{\pi}{4}(x_1^2 - 1) + O((x_1^2 - 1)^2) \right) \left(-\pi + \frac{\pi}{4}(x_2^2 - 1) + O((x_2^2 - 1)^2) \right),$$

$$h = \frac{(x_1^2 - 1)(x_2^2 - 1)}{4}, \quad \eta = \sqrt{2}\pi.$$

Now we investigate the approximation error between $\mathcal{N}_{2,M,\text{ReLU},h}$ and $H_{y^\alpha}^{1,b}(\mathcal{D})$. It suffices to consider a special subset $\mathcal{S}_{2,M,\text{ReLU},h}$ given by

$$(3.11) \quad \mathcal{S}_{2,M,\text{ReLU},h} = \{ \hat{u} \mid \hat{u}(z) = \hat{v}(z)h(x)e^{-\eta y}, \quad \forall \hat{v} \in \mathcal{F}_{2,M,\text{ReLU}} \}.$$

Note in $\mathcal{S}_{2,M,\text{ReLU},h}$ only the parameters of \hat{v} are free and trainable, while η is fixed. Clearly, $\mathcal{S}_{2,M,\text{ReLU},h}$ is a subset of $\mathcal{N}_{2,M,\text{ReLU},h}$. The following theorem and proof are referred to the result in [23] for deep Ritz method in a bounded domain.

THEOREM 3.1. *Let $u \in H_{y^\alpha}^{1,b}(\mathcal{D})$. If Assumption 3.1 is true with $|h| \leq 1$ and $|\nabla h| \leq 1$, then there exists some $\hat{u} \in \mathcal{S}_{2,M,\text{ReLU},h}$ such that*

$$(3.12) \quad \|\hat{u} - u\|_{H_{y^\alpha}^{1,b}(\mathcal{D})} \leq C(\Omega, \eta) M^{-\frac{1}{2}} \|v\|_{\mathcal{B}},$$

with

$$(3.13) \quad C(\Omega, \eta) = 2^{\frac{3}{2}} |\Omega|^{\frac{1}{2}} \left[\frac{R_\Omega^2 + 1}{\alpha + 1} + \frac{1}{\alpha + 3} + \frac{(4\eta^3 + 2\eta^2)(R_\Omega^2 + 1) + 4\eta^3 + 6\eta^2 + 6\eta + 3}{8\eta^4} e^{-2\eta} \right]^{\frac{1}{2}},$$

where $R_\Omega := \max(\sup_{x \in \Omega} |x|, 1)$.

Proof. By the definition of the Barron norm, there exists some probability measure π such that $f_\pi = v$ a.e. and $\mathbb{E}_\pi [a^2(|b| + |c|)^2] \leq 2\|v\|_{\mathcal{B}}^2$. For all $(a, b, c) \in \mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{R}$, using the facts $|\sigma(t)| \leq |t|$ and $|\sigma'(t)| = \chi_{t \geq 0}$ we have

$$(3.14) \quad \begin{aligned} \|a\sigma(b^T z + c)\|_{1, e^{-2\eta y} y^\alpha, \mathcal{D}}^2 &= \int_{\mathcal{D}} [|a\sigma(b^T z + c)|^2 + |\nabla(a\sigma(b^T z + c))|^2] e^{-2\eta y} y^\alpha dz \\ &\leq \int_{\mathcal{D}} [a^2(b^T z + c)^2 + a^2 \chi_{b^T z + c \geq 0} |\nabla(b^T z + c)|^2] e^{-2\eta y} y^\alpha dz \\ &\leq \int_{\mathcal{D}} [a^2(|b||z| + |c|)^2 + a^2|b|^2] e^{-2\eta y} y^\alpha dz. \end{aligned}$$

Since $|z| = (|x|^2 + y^2)^{\frac{1}{2}} \leq (R_\Omega^2 + y^2)^{\frac{1}{2}}$, we have

$$(3.15) \quad \begin{aligned} &\int_{\mathcal{D}} [a^2(|b||z| + |c|)^2 + a^2|b|^2] e^{-2\eta y} y^\alpha dz \\ &\leq \int_{\mathcal{D}} [(R_\Omega^2 + y^2)a^2(|b| + |c|)^2 + a^2|b|^2] e^{-2\eta y} y^\alpha dz \\ &\leq a^2(|b| + |c|)^2 \int_{\mathcal{D}} (R_\Omega^2 + y^2 + 1) e^{-2\eta y} y^\alpha dz \\ &\leq a^2(|b| + |c|)^2 |\Omega| \int_0^\infty (R_\Omega^2 + y^2 + 1) e^{-2\eta y} y^\alpha dy := a^2(|b| + |c|)^2 |\Omega| \cdot I_1. \end{aligned}$$

While I_1 is bounded above since

$$(3.16) \quad \begin{aligned} I_1 &\leq \int_0^1 (R_\Omega^2 + y^2 + 1) y^\alpha dy + \int_1^\infty (R_\Omega^2 + y^2 + 1) e^{-2\eta y} y^\alpha dy \\ &= \left(\frac{R_\Omega^2 + 1}{\alpha + 1} + \frac{1}{\alpha + 3} \right) + \frac{(4\eta^3 + 2\eta^2)(R_\Omega^2 + 1) + 4\eta^3 + 6\eta^2 + 6\eta + 3}{8\eta^4} e^{-2\eta} := I_2. \end{aligned}$$

Combining (3.14),(3.15),(3.16) we have

$$(3.17) \quad \mathbb{E}_\pi \left[\|a\sigma(b^T z + c)\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right] \leq \mathbb{E}_\pi \left[a^2(|b| + |c|)^2 \right] |\Omega| \cdot I_2 \leq 2I_2 |\Omega| \cdot \|v\|_{\mathcal{B}}^2.$$

On the other hand, the mapping

$$\mathbb{R} \times \mathbb{R}^{d+1} \times \mathbb{R}, \quad (a, b, c) \rightarrow a\sigma(b^T z + c)$$

is continuous and hence Bochner measurable. Also, (3.17) leads to

$$\mathbb{E}_\pi \left[\|a\sigma(b^T z + c)\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right] \leq \left(\mathbb{E}_\pi \left[\|a\sigma(b^T z + c)\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right] \right)^{\frac{1}{2}} < \infty,$$

which implies the integral $\int_{\mathcal{D}} a\sigma(b^T z + c)\pi(da, db, dc)$ is a Bochner integral.

We note the fact that if ξ_1, \dots, ξ_M are independent samples from a random variable ξ , then

$$\begin{aligned} \mathbb{E} \left(\frac{1}{M} \sum_{i=1}^M \xi_i - \mathbb{E}\xi \right)^2 &= \mathbb{E} \left(\frac{1}{M} \sum_{i=1}^M (\xi_i - \mathbb{E}\xi) \right)^2 \\ &= \frac{1}{M^2} \left(\sum_{i=1}^M \mathbb{E}(\xi_i - \mathbb{E}\xi)^2 + \sum_{1 \leq i < j \leq M} \mathbb{E}(\xi_i - \mathbb{E}\xi) \cdot \mathbb{E}(\xi_j - \mathbb{E}\xi) \right) = \frac{1}{M} \sum_{i=1}^M \mathbb{E}(\xi_i - \mathbb{E}\xi)^2 \\ &= \frac{1}{M} \mathbb{E}\xi^2 - \frac{1}{M} (\mathbb{E}\xi)^2 \leq \frac{1}{M} \mathbb{E}\xi^2. \end{aligned}$$

By similar argument, for independent samples $\{(a_i, b_i, c_i)\}$ from π , we have

$$\mathbb{E}_{\pi^M} \left[\left\| \frac{1}{M} \sum_{i=1}^M a_i \sigma(b_i^T z + c_i) - f_\pi(z) \right\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right] \leq \frac{1}{M} \mathbb{E}_\pi \left[\|a\sigma(b^T z + c)\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right].$$

In particular, there exists $\{(a_i, b_i, c_i)\}$ such that

$$(3.18) \quad \left\| \frac{1}{M} \sum_{i=1}^M a_i \sigma(b_i^T z + c_i) - f_\pi(z) \right\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \leq \frac{1}{M} \mathbb{E}_\pi \left[\|a\sigma(b^T z + c)\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \right].$$

Let $\hat{v} = \frac{1}{M} \sum_{i=1}^M a_i \sigma(b_i^T z + c_i)$, combining (3.17) and (3.18) have obtain

$$(3.19) \quad \|\hat{v} - v\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2 \leq 2I_2 |\Omega| M^{-1} \cdot \|v\|_{\mathcal{B}}^2$$

Finally, let $\hat{u} = \hat{v}(z)h(x)e^{-\eta y}$ and denote $\bar{v} := \hat{v} - v$, then

$$\begin{aligned} &\|\hat{u} - u\|_{H_{y^\alpha}^{1,b}(\mathcal{D})}^2 \\ &= \|\nabla(\bar{v}(z)h(x)e^{-\eta y})\|_{y^\alpha, \mathcal{D}}^2 \\ &= \int_{\mathcal{D}} (|h(x)\nabla_x \bar{v}(z) + \bar{v}(z)\nabla_x h(x)|^2 + |\partial_y \bar{v}(z) - \bar{v}(z)|^2 h^2(x)) e^{-2\eta y} y^\alpha dz \\ (3.20) \quad &\leq 4 \int_{\mathcal{D}} (|\nabla \bar{v}|^2 + |\bar{v}|^2) e^{-2\eta y} y^\alpha dz = 4\|\hat{v} - v\|_{1,e^{-2\eta y} y^\alpha, \mathcal{D}}^2. \end{aligned}$$

The inequality (3.12) directly follows (3.19) and (3.20). \square

Theorem 3.1 provides a fractional equation dimension-independent approximation error bound for neural networks in $\mathcal{S}_{2,M,\text{ReLU},h}$ given that the target function u has a Barron component. This is a desired property since it avoids the curse of dimensionality. Since $\mathcal{S}_{2,M,\text{ReLU},h} \subset \mathcal{N}_{2,M,\text{ReLU},h}$, Theorem 3.1 also holds for $\mathcal{N}_{2,M,\text{ReLU},h}$.

4. Ritz method and Error Estimation. The extended problem (1.4) can be solved practically by Ritz method. Thanks to the approximation property of the neural network class discussed in Section 3.3, we can replace the hypothesis space $H_{y^\alpha}^{1,b}(\mathcal{D})$ with $\mathcal{N}_{L,M,\sigma,h}$ in the minimization (2.22), obtaining

$$(4.1) \quad \min_{\hat{w} \in \mathcal{N}_{L,M,\sigma,h}} \mathcal{I}[\hat{w}].$$

Then the solution of (4.1) will be an approximation to the solution of (2.22). Same as in Section 3.3, we will estimate the solution error for the two-layer ReLU networks; namely, we consider the case that $L = 2$ and $\sigma(t) = \max(0, t)$. The final error of the original problem (1.1) will thereafter be presented.

4.1. Error estimation. Let $u^* \in H_{y^\alpha}^{1,b}(\mathcal{D})$ be a minimizer of (2.22); namely,

$$(4.2) \quad \mathcal{I}[u^*] = \min_{w \in H_{y^\alpha}^{1,b}(\mathcal{D})} \mathcal{I}[w]$$

Given that Assumption 3.1 holds for u^* with a factorization $u^*(z) = v^*(z)h^*(x)e^{-\eta^*y}$, let $\hat{u}^* \in \mathcal{N}_{2,M,\text{ReLU},h^*}$ be a minimizer of (4.1); namely,

$$(4.3) \quad \mathcal{I}[\hat{u}^*] = \min_{\hat{w} \in \mathcal{N}_{2,M,\text{ReLU},h^*}} \mathcal{I}[\hat{w}].$$

We first introduce the following Céa Lemma [22].

PROPOSITION 4.1. *Let X be a Hilbert space, $V \subset X$ any subset and $a : X \times X \rightarrow \mathbb{R}$ a symmetric, continuous and α -coercive bilinear form. For $f \in X'$ define the quadratic energy $E_f(u) := \frac{1}{2}a(u, u) - f(u)$ and denote its unique minimizer by u_f . Then for every $v \in V$ it holds that*

$$(4.4) \quad \|v - u_f\|_X \leq \sqrt{\alpha^{-1} \left(2 \left(E_f(v) - \inf_{\tilde{v} \in V} E_f(\tilde{v}) \right) + \inf_{\tilde{v} \in V} (\tilde{v} - u_f, \tilde{v} - u_f) \right)}.$$

It is trivial to show that $(\nabla \cdot, \nabla \cdot)_{y^\alpha, \mathcal{D}}$ is a symmetric, continuous and 1-coercive bilinear form. Therefore by Lemma 4.1 and Theorem 3.1 we have

$$(4.5) \quad \begin{aligned} \|\hat{u}^* - u^*\|_{H_{y^\alpha}^{1,b}(\mathcal{D})} &\leq \sqrt{\inf_{\hat{u} \in \mathcal{N}_{2,M,\text{ReLU},h^*}} (\nabla(\hat{u} - u^*), \nabla(\hat{u} - u^*))_{y^\alpha, \mathcal{D}}} \\ &\leq \inf_{\hat{u} \in \mathcal{N}_{2,M,\text{ReLU},h^*}} \|\hat{u} - u^*\|_{H_{y^\alpha}^{1,b}(\mathcal{D})} \leq C(\Omega, \eta^*) M^{-\frac{1}{2}} \|v^*\|_{\mathcal{B}}. \end{aligned}$$

To derive the error for the original problem (1.1), we introduce the following trace theorem [24],

PROPOSITION 4.2. *The trace operator tr defined in (2.8) satisfies $\text{tr}\{H_{y^\alpha}^{1,b}(\mathcal{D})\} = \mathbb{H}^s(\Omega)$, and*

$$(4.6) \quad \|\text{tr}\{u\}\|_{\mathbb{H}^s(\Omega)} \leq C \|u\|_{H_{y^\alpha}^{1,b}(\mathcal{D})}, \quad \forall u \in H_{y^\alpha}^{1,b}(\mathcal{D}),$$

where C is a constant only depending on Ω and s , and the space $\mathbb{H}^s(\Omega)$ is defined as

$$(4.7) \quad \mathbb{H}^s(\Omega) = \left\{ u = \sum_{n=1}^{\infty} \tilde{u}_n \phi_n \in L^2(\Omega) : \|u\|_{\mathbb{H}^s} = \sum_{n=1}^{\infty} \lambda_n^s |\tilde{u}_n|^2 < \infty \right\},$$

with (λ_n, ϕ_n) being all the eigenvalues and orthonormal eigenfunctions of (1.2). Especially, $\mathbb{H}^s(\Omega)$ can be characterized by

$$(4.8) \quad \mathbb{H}^s(\Omega)u = \begin{cases} H^s(\Omega), & s \in (0, \frac{1}{2}), \\ H_{00}^{\frac{1}{2}}(\Omega), & s = \frac{1}{2}, \\ H_0^s(\Omega), & s \in (\frac{1}{2}, 1), \end{cases}$$

where

$$(4.9) \quad H_{00}^{\frac{1}{2}}(\Omega) = \left\{ u \in H^{\frac{1}{2}}(\Omega) : \int_{\Omega} \frac{|u(x)|^2}{\text{dist}(x, \partial\Omega)} dx < \infty \right\}.$$

Recall that the solution of original problem (1.1) is exactly the trace of the solution of the extended problem (1.4). Also, Theorem 2.1 shows the equivalence between the extended problem (1.4) and the minimization problem (2.22). Therefore, combining (4.5) and (4.6) leads to the following error estimate for the original solution.

THEOREM 4.3. *Given that the solution u^* of (2.22) satisfies Assumption 3.1, say $u^*(z) = v^*(z)h^*(x)e^{-\eta^*y}$. Let $U^*(x)$ be the solution of (1.1) and $\hat{u}^*(x, y)$ be the solution of (4.1) with $L = 2$, $h = h^*$ and $\sigma(\cdot) = \max(0, \cdot)$. Then*

$$(4.10) \quad \|\hat{u}^*(x, 0) - U^*(x)\|_{\mathbb{H}^s(\Omega)} \leq CM^{-\frac{1}{2}}\|v^*\|_{\mathcal{B}},$$

where C is a constant only depending on Ω , s and η^* .

For our network-based method, we obtain the solution error $O(M^{-1/2})$ provided that the true solution has a Barron component. This order $-1/2$ is consistent with that of the deep Ritz method solving regular Laplacian equations [23] proved under the similar Barron framework. Moreover, the proposed method can be compared with existing methods solving fractional Laplacian [24, 11]. In the early work, finite elements or Laguerre functions are used to approximate the solution, having error orders of $O(N_p^{-\nu/d})$, where ν characterizes the regularity of the true solution and N_p is the number of free parameters. If d is relatively low, say $d < 2\nu$, then $-\nu/d < -1/2$ and hence the early methods converges faster. Otherwise, the network-based methods outperform the existing methods in the error order. This fact implies that neural networks are advantageous over other classical structures in the high-dimensional approximation.

4.2. Implementation. In the proposed method, we solve the optimization (4.1), finding a minimizer of $\mathcal{I}[\cdot]$ in the hypothesis space $\mathcal{N}_{L,M,\sigma,h}$. Note that

$$(4.11) \quad \mathcal{I}[\hat{\phi}(x, y)] = \frac{1}{2} \int_0^\infty y^\alpha \left(\int_{\Omega} |\nabla \hat{\phi}(x, y)|^2 dx \right) dy - d_s \int_{\Omega} f(x) \hat{\phi}(x, 0) dx.$$

In practice, numerical quadrature is required for the integrals in (4.11). Since d might be moderately large, for the integral in terms of x over Ω , one choice is using Monte Carlo-type quadrature.

Specifically, we prescribe a set of N quadrature nodes $\mathcal{T} = \{x_n\}_{n=1}^N$ which are uniformly distributed in Ω , then for each $y \in (0, \infty)$,

$$(4.12) \quad \int_{\Omega} |\nabla \hat{\phi}(x, y)|^2 dx \approx \frac{|\Omega|}{N} \sum_{n=1}^N |\nabla \hat{\phi}(x_n, y)|^2, \quad \int_{\Omega} f(x) \hat{\phi}(x, 0) dx \approx \frac{|\Omega|}{N} \sum_{n=1}^N f(x_n) \hat{\phi}(x_n, 0).$$

For the integral in terms of y over $(0, \infty)$, specific quadrature rule is needed due to the singularity at $y = 0$. It is shown in [4, 6] that the infinite integral involving fractional operators can be effectively computed by sinc quadrature. More precisely, we use the change of variable $y = e^\mu$ so that

$$(4.13) \quad \int_0^\infty y^\alpha g(y) dy = \int_{-\infty}^\infty e^{(\alpha+1)\mu} g(e^\mu) d\mu$$

for all $g \in L_{y^\alpha}^1(0, \infty)$. Given $\bar{h} > 0$, let

$$(4.14) \quad N_+ := \lceil \frac{\pi^2}{4s\bar{h}^2} \rceil, \quad N_- := \lceil \frac{\pi^2}{4(1-s)\bar{h}^2} \rceil, \quad \mu_m := m\bar{h},$$

then (4.13) is evaluated by quadrature nodes $\{\mu_m\}_{m=-N_-, \dots, N_+}$ and uniform weights \bar{h} , namely,

$$(4.15) \quad \int_{-\infty}^\infty e^{(\alpha+1)\mu} g(e^\mu) d\mu \approx \bar{h} \sum_{m=-N_-}^{N_+} e^{(\alpha+1)\mu_m} g(e^{\mu_m}).$$

Combining (4.12) and (4.15) we have an approximate functional of \mathcal{I} given by

$$(4.16) \quad \mathcal{I}_{\mathcal{T}, \bar{h}}[\hat{\phi}(x, y)] = \frac{|\Omega|}{N} \left(\bar{h} \sum_{m=-N_-}^{N_+} e^{(\alpha+1)\mu_m} \sum_{n=1}^N |\nabla \hat{\phi}(x_n, e^{\mu_m})|^2 - d_s \sum_{n=1}^N f(x_n) \hat{\phi}(x_n, 0) \right).$$

Practically, we solve the following optimization,

$$(4.17) \quad \min_{\hat{\phi} \in \mathcal{N}_{L, M, \sigma, h}} \mathcal{I}_{\mathcal{T}, \bar{h}}[\hat{\phi}],$$

whose solution can be regarded as a practical numerical solution of (2.22).

5. Numerical Experiments.

5.1. The setting. The proposed method is tested by numerical experiments in this section. Deep learning techniques are utilized to solve the minimization (4.17). The overall setting is summarized as follows.

- *Environment.* The experiment is performed in Python environment. PyTorch library with CUDA toolkit is utilized for neural network implementation and GPU-based parallel computing. The codes can be simply implemented on a desktop.
- *Optimizer and hyper-parameters.* The network-based optimization (4.17) is solved by the stochastic gradient descent (SGD) from PyTorch library. The SGD is implemented for totally 5000 epochs, with adaptively decaying learning rates.

- *Network setting.* The special network structure (3.5) is adopted. We choose σ as the ReLU function. The parameters in θ' and θ'' are initialized by

$$(5.1) \quad a, W_l, b_l \sim U(-\sqrt{M}, \sqrt{M}), \quad l = 1, \dots, L.$$

And γ', γ'' are initialized as 0.5. All these parameters are trained in the learning process.

- *Numerical quadrature.* For quadrature (4.12) over Ω , we adopt the quasi-Monte Carlo with Halton sequence. For the cases of $d = 3$ and 10, totally $N = 10^5$ and 5×10^5 sample points are used in the Monte Carlo quadrature, separated as 4 and 10 batches in the SGD, respectively. The sinc quadrature parameter \bar{h} is set as $1/3$.
- *Testing set and error evaluation.* We generate a testing set \mathcal{X} consisting of 10^4 random points uniformly distributed in Ω for error evaluation. Suppose $\hat{\phi}(x, y)$ is the neural network obtained by our method and $U(x)$ is the true solution of the original fractional Laplacian problem (1.1), then the following relative ℓ^2 error will be computed,

$$(5.2) \quad e_{\ell^2}(\mathcal{X}) := \left(\sum_{x \in \mathcal{X}} |\hat{\phi}(x, 0) - U(x)|^2 / \sum_{x \in \mathcal{X}} |U(x)|^2 \right)^{\frac{1}{2}}.$$

5.2. A model problem. In this example, we solve the following problem with an explicit solution to test the accuracy of the proposed method,

$$(5.3) \quad \begin{cases} (-\Delta)^s u(x) = (d\pi^2)^s \prod_{i=1}^d \sin(\pi x_i), & \forall x := [x_1, \dots, x_d] \in \Omega := (-1, 1)^d, \\ u(x) = 0, & \forall x \in \partial\Omega, \end{cases}$$

whose true solution is given by $u(x) = \prod_{i=1}^d \sin(\pi x_i)$. We specify the boundary function $h(x) = \prod_{i=1}^d (1 - x_i^2)$ in the network architecture (3.5).

5.2.1. Network size test. First, we solve the problem (5.3) with $s = 0.5$ using special network (3.5) of various network sizes. The proposed method is implemented with network depth $L = 2, 3$ and width $M = 25, 50, 100, 200$. The errors of the numerical solutions are listed in Table 5.1 as well as the total running time. We also computed the numerical error orders with respect to M . Note that most of the running time is occupied by training the networks, and very little is for the testing process (computing the errors).

From the table, it is observed that using larger sizes improves the accuracy at the expense of extra running time. For a fixed width M , the obtained error is reduced by more than half from $L = 2$ to $L = 3$. For a fixed depth L , the numerical order with respect to M is roughly around the theoretical order $-1/2$ proved in Theorem 4.3, and the deviation is probably due to the stochasticity and capability of the learning algorithm. Overall, the size pair $(L, M) = (3, 200)$ obtains the most accurate solution; hence we continue using it in the following tests.

5.2.2. Comparison of structures. Next, this problem is solved for $d = 3$ with $s = 0.1, 0.3, 0.5, 0.7, 0.9$. We test both the proposed special structure (3.5) and the following simple FNN structure for a comparison,

$$(5.4) \quad \hat{\phi}(x, y) = \hat{\phi}'(x, y)h(x)e^{-\frac{1}{2}y},$$

where $\hat{\phi}'$ is a generic FNN. Note in (5.4), the power of the exponent is fixed with $-1/2$ instead of a trainable parameter, and no singular term is introduced. By this design, the comparison can reveal

M	$L = 2$			$L = 3$		
	Error	Order	Running Time	Error	Order	Running Time
25	8.252×10^{-2}	N.A.	7.8×10^3	2.410×10^{-2}	N.A.	8.0×10^3
50	4.337×10^{-2}	-0.93	7.7×10^3	2.071×10^{-2}	-0.22	8.0×10^3
100	3.522×10^{-2}	-0.3	7.9×10^3	1.371×10^{-2}	-0.59	8.6×10^3
200	2.529×10^{-2}	-0.48	8.6×10^3	8.424×10^{-3}	-0.7	1.3×10^4

Table 5.1: Errors $e_{\ell^2}(\mathcal{X})$, error orders with respect to M and running time (seconds) for various L and M when $d = 3$.

the advantages of using trainable exponential decaying powers and the special singular term. For both structures, we set $L = 3$ and $M = 200$ for the involved FNN.

The error curves versus epochs of the SGD are shown in Figure 5.1, and the errors of the finally obtained solutions are listed in Table 5.2. It is observed that for each structure, the smallest error is obtained when $s = 0.5$, while larger errors are obtained when s is close to 0 or 1. This is natural since the true solution has no singularity if $s = 0.5$ and has higher singularity if s approaches to 0 or 1. Moreover, from the comparison, it is clear that the special structure outperforms the simple one in obtaining smaller errors. We also remark that the time cost of the simple structure is only slightly less than the special structure.

5.2.3. High-dimensional simulation. Finally, we conduct a high-dimensional test by solving the model problem (5.3) of $d = 10$ and $s = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$. We continue using the special structure of $L = 3$ and $M = 200$ for approximation. The error curves and final errors are shown in Figure 5.2 and Table 5.3. We observe that the proposed method is still effective for high dimensions, obtaining the errors $O(10^{-2})$. While the smallest error is obtained between $s = 0.4$ and 0.5 .

To the best of our knowledge, our method is the first successful attempt at 10-D spectral fractional Laplacian equation. The existing methods [24, 11] using finite elements or Laguerre functions show their effectiveness in low-dimensional problems ($d = 1, 2$), while they are incapable of high-dimensional cases due to the limitation of storage. Therefore, we do not conduct a numerical comparison in this work.

6. Conclusion. In summary, we develop a novel deep learning-based method to solve the spectral fractional Laplacian equation numerically. First, we reformulate the fractional equation as a regular partial differential equation of one more dimension by the Caffarelli-Silvestre extension. Next, we transform the extended problem to an equivalent minimal functional problem, and characterize the space of the weak solutions. To deal with the possible high dimensions, we employ FNNs to construct a special approximate class of the solution space, by which the approximate solution has consistent properties of the true solution. We then solve the minimization in the approximate class. In theory, we studied the approximation error of the special class under Barron-type hypothesis, and thereafter derive the solution error of this method. Finally, the effectiveness of the proposed deep Ritz method is illustrated in high-dimensional simulations.

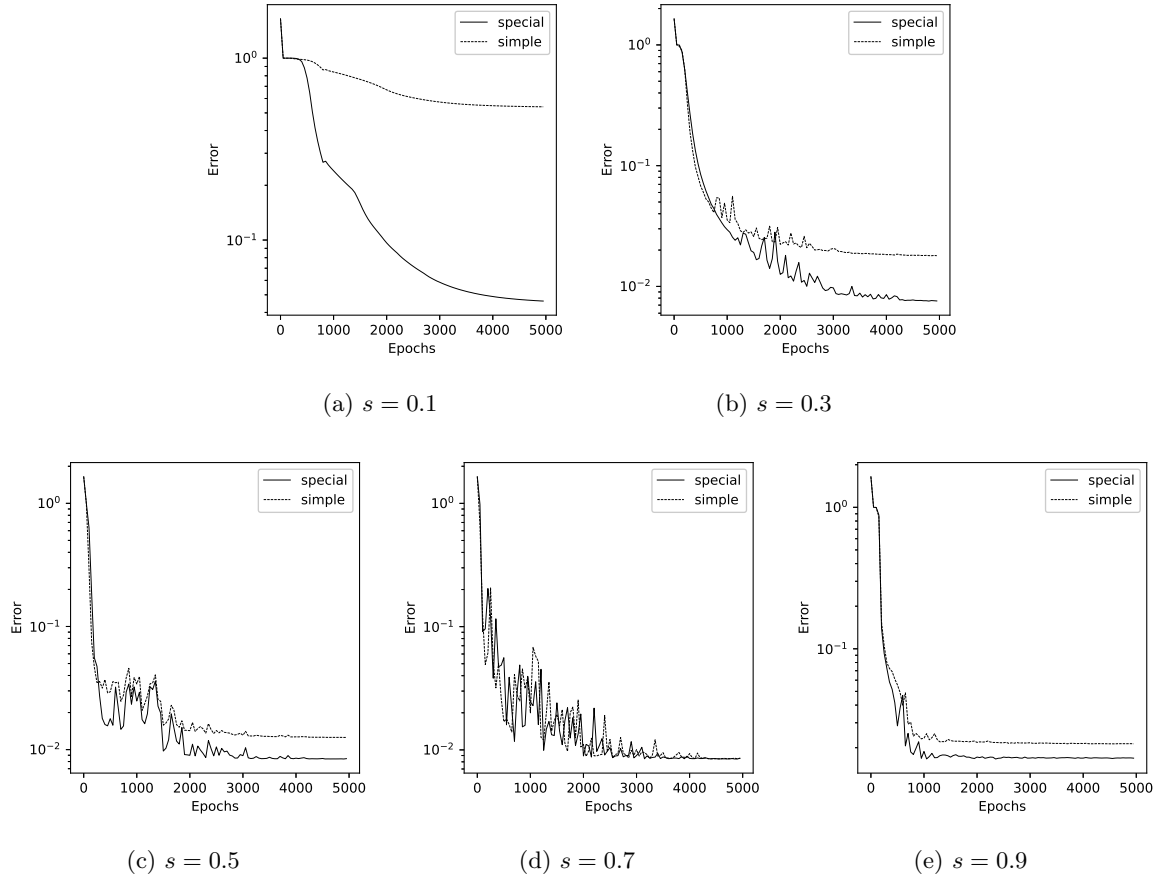


Fig. 5.1: Errors $e_{\ell^2}(\mathcal{X})$ versus epochs for various s using the special or simple structure when $d = 3$.

s	Special structure in (3.5)		Simple structure in (5.4)	
	Error	Running Time	Error	Running Time
0.1	4.619×10^{-2}	8.7×10^3	5.402×10^{-1}	8.5×10^3
0.3	7.576×10^{-3}	9.9×10^3	1.800×10^{-2}	9.6×10^3
0.5	8.424×10^{-3}	1.3×10^4	1.257×10^{-2}	1.1×10^4
0.7	8.476×10^{-3}	1.8×10^4	8.498×10^{-3}	1.5×10^4
0.9	1.691×10^{-2}	2.5×10^4	2.134×10^{-2}	2.3×10^4

Table 5.2: Errors $e_{\ell^2}(\mathcal{X})$ and running time (seconds) for various s using the special or simple structure when $d = 3$.

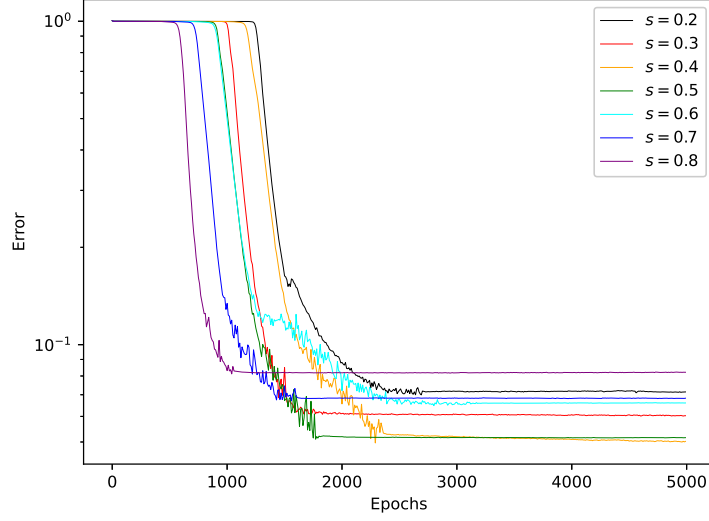


Fig. 5.2: Errors $e_{\ell^2}(\mathcal{X})$ versus epochs for various s when $d = 10$.

s	Error	Running Time
0.2	7.134×10^{-2}	2.4×10^4
0.3	6.029×10^{-2}	2.5×10^4
0.4	5.012×10^{-2}	2.7×10^4
0.5	5.158×10^{-2}	2.9×10^4
0.6	6.603×10^{-2}	3.4×10^4
0.7	6.824×10^{-2}	4.1×10^4
0.8	8.215×10^{-2}	5.6×10^4

Table 5.3: Errors $e_{\ell^2}(\mathcal{X})$ and running time (seconds) for various s when $d = 10$.

In this work, we consider the error between the minimizer of the energy functional \mathcal{I} in the approximation class and the true solution of the original problem. However, practically one needs to use numerical quadrature to compute \mathcal{I} , which brings extra errors. Consequently, future work may include the error analysis for the minimizer of the empirical loss function $\mathcal{I}_{\mathcal{T}, \bar{h}}$ (4.16) discretized from \mathcal{I} by the Monte Carlo method and sinc quadrature. More precisely, let u^* be the true solution of (1.4) and

$$(6.1) \quad \hat{u} = \operatorname{argmin}_{w \in \mathcal{N}} \mathcal{I}[w], \quad \hat{u}_{\mathcal{T}, \bar{h}} = \operatorname{argmin}_{w \in \mathcal{N}} \mathcal{I}_{\mathcal{T}, \bar{h}}[w]$$

for some DNN-based hypothesis space \mathcal{N} , then

$$(6.2) \quad \|u^* - \hat{u}_{\mathcal{T}, \bar{h}}\| \leq \|u^* - \hat{u}\| + \|\hat{u} - \hat{u}_{\mathcal{T}, \bar{h}}\|.$$

Note that $\|u^* - \hat{u}\|$ has been investigated, it suffices to consider $\|\hat{u} - \hat{u}_{\mathcal{T},\bar{h}}\|$. Suppose \mathcal{I} has a bounded inverse, then

$$\begin{aligned}
 (6.3) \quad \|\hat{u} - \hat{u}_{\mathcal{T},\bar{h}}\| &\leq \|\mathcal{I}^{-1}\| \|\mathcal{I}[\hat{u}] - \mathcal{I}[\hat{u}_{\mathcal{T},\bar{h}}]\| \leq \|\mathcal{I}^{-1}\| (\mathcal{I}[\hat{u}_{\mathcal{T},\bar{h}}] - \mathcal{I}[\hat{u}]) \\
 &\leq \|\mathcal{I}^{-1}\| (\mathcal{I}[\hat{u}_{\mathcal{T},\bar{h}}] - \mathcal{I}[\hat{u}] + \mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}_{\mathcal{T},\bar{h}}] - \mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}_{\mathcal{T},\bar{h}}]) \\
 &\leq \|\mathcal{I}^{-1}\| [(\mathcal{I}[\hat{u}_{\mathcal{T},\bar{h}}] - \mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}_{\mathcal{T},\bar{h}}]) + (\mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}] - \mathcal{I}[\hat{u}])],
 \end{aligned}$$

where we use the facts $\mathcal{I}[\hat{u}] \leq \mathcal{I}[\hat{u}_{\mathcal{T},\bar{h}}]$ and $\mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}_{\mathcal{T},\bar{h}}] \leq \mathcal{I}_{\mathcal{T},\bar{h}}[\hat{u}]$. On the right hand side of (6.3), the first term describes the generalization error of the empirical loss minimization over the hypothesis space, and the second term characterizes the bias coming from the numerical quadrature of the integrals. In [19], related analysis is conducted for Poisson equation and Schrödinger equation using Rademacher complexity, but it is only for the Monte-Carlo quadrature. It is promising and challenging to do similar analysis for our method on fractional Laplacian equations, which involves both the stochastic Monte-Carlo method and the deterministic sinc quadrature for approximation.

REFERENCES

- [1] G. Acosta and J. P. Borthagaray. A fractional Laplace equation: regularity of solutions and finite element approximations. *SIAM J Numer Anal*, 55:472–495, 2017.
- [2] A. R. Barron. Neural net approximation. In *Proceedings of the 7th Yale Workshop on Adaptive and Learning Systems*, 1992.
- [3] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory*, 39(3):930–945, 1993.
- [4] A. Bonito, J. P. Borthagaray, R. H. Nochetto, E. Otarola, and A. J. Salgado. Numerical methods for fractional diffusion. *arXiv e-prints*, arXiv:1707.01566, 2017.
- [5] A. Bonito, W. Lei, and J. E. Pasciak. Numerical approximation of the integral fractional Laplacian. *Numer Math*, 142:235–278, 2019.
- [6] A. Bonito, W. Lei, and J. E. Pasciak. On sinc quadrature approximations of fractional powers of regularly accretive operators. *J Numer Math*, 27(2):57–68, 2019.
- [7] L. Caffarelli and L. Silvestre. An extension problem related to the fractional Laplacian. *Comm Partial Differential Equations*, 32(8):1245–1260, 2007.
- [8] A. Capella, J. Dávila, L. Dupaigne, and Y. Sire. Regularity of radial extremal solutions for some non-local semilinear equations. *Comm Partial Differential Equations*, 36(8):1353–1384, 2011.
- [9] A. Caragea, P. Petersen, and F. Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *arXiv e-prints*, arXiv:2011.09363, 2020.
- [10] B. A. Carreras, V. E. Lynch, and G. M. Zaslavsky. Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence model. *Phys Plasmas*, 8:5096–5103, 2001.
- [11] S. Chen and J. Shen. An efficient and accurate numerical method for the spectral fractional Laplacian equation. *J Sci Comput*, 82(17), 2020.
- [12] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential lévy models. *SIAM J Numer Anal*, 43:1596–1626, 2005.
- [13] W. E, C. Ma, S. Wojtowytsch, and L. Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don’t. *arXiv e-prints*, arXiv:2009.10713, 2020.
- [14] W. E, C. Ma, and L. Wu. The Barron space and the flow-induced function spaces for neural network models. *arXiv e-prints*, arXiv:1906.08039, 2019.
- [15] W. E and S. Wojtowytsch. Representation formulas and pointwise properties for Barron functions. *arXiv e-prints*, arXiv:2006.05982, 2020.
- [16] B. Jourdain, S. Méléard, and W. A. Woyczynski. A probabilistic approach for nonlinear equations involving the fractional Laplacian and a singular operator. *Potential Anal*, 23(1):55–81, 2005.
- [17] J. M. Klusowski and A. R. Barron. Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Trans. Inform. Theory*, 64(12):7649–7656, 2018.
- [18] N. Laskin. Fractional quantum mechanics and lévy path integrals. *Phys Lett A*, 268:298–305, 2000.

- [19] J. Lu, Y. Lu, and M. Wang. A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic equations. *arXiv e-prints*, arXiv:2101.01708, 2021.
- [20] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. *arXiv e-prints*, arXiv:2001.03040, 2020.
- [21] J. Lund and K. L. Bowers. *Sinc methods for quadrature and differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
- [22] K. S. Miller and S. G. Samko. Completely monotonic functions. *Integr Transf Spec F*, 12(4):389–402, 2001.
- [23] Johannes Müller and Marius Zeinhofer. Error estimates for the variational training of neural networks with boundary penalty. *arXiv e-prints*, arXiv:2103.01007, 2021.
- [24] R. H. Nochetto, E. Otárola, and A. J. Salgado. A PDE approach to fractional diffusion in general domains: a priori error analysis. *Found Comput Math*, 15:733–791, 2015.
- [25] Z. Shen, H. Yang, and S. Zhang. Deep network with approximation error being reciprocal of width to power of square root of depth. *Neural Comput*, 33(4):1005–1036, 2021.
- [26] M. F. Shlesinger, B. J. West, and J. Klafter. Lévy dynamics of enhanced diffusion: application to turbulence. *Phys Rev Lett*, 58:1100–1103, 1987.
- [27] J. W. Siegel and J. Xu. Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321, 2020.
- [28] J. W. Siegel and J. Xu. High-order approximation rates for neural networks with ReLU^k activation functions. *arXiv e-prints*, arXiv:2012.07205, 2020.
- [29] P. Tankov. *Financial modelling with jump processes*, volume 2. CRC press, 2003.
- [30] V. V. Uchaikin. Nonlocal models of cosmic ray transport in the galaxy. *J Appl Math Phys*, 3:187–200, 2015.
- [31] J. L. Vázquez. Recent progress in the theory of nonlinear diffusion with fractional Laplacian operators. *Discrete Contin Dyn*, 7:857–885, 2014.