ROLE OF NONLOCAL OPERATORS IN INVERSE AND DEEP LEARNING PROBLEMS

by

Ratna Khatri
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Mathematics

Committee:

Dr. Harbir Antil, Dissertation Director

Dr. Mahamadi Warma, Committee Member

Dr. Igor Griva, Committee Member

Dr. Carlos Rautenberg, Committee Member

Dr. Zichao (Wendy) Di, Committee Member

Dr. David Walnut, Department Chairperson

Dr. Donna M. Fox, Associate Dean,
Office of Student Affairs & Special
Programs, College of Science

Dr. Ali Andalibi, Dean, College of Science

Date:

Spring Semester 2020
George Mason University
Fairfax, VA

Role of Nonlocal Operators in Inverse and Deep Learning Problems

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Ratna Khatri
Bachelor of Science
George Mason University, 2016

Director: Dr. Harbir Antil,
Director, Center for Mathematics and Artificial Intelligence (CMAI)
Associate Professor, Department of Mathematical Sciences

Spring Semester 2020
George Mason University
Fairfax, VA

ॐ

Dedication

For my family,

Mansukh Ram Khatri and Vasudha Khatri
Alpna Khatri
Manisha Khatri
Daksha Khatri

a thousand times over...

Acknowledgments

First and foremost, I would like thank my advisor Dr. Harbir Antil for his unwavering support. His leadership and enthusiastic spirit of making things happen has been immensely inspiring. His intellectual prowess, his vision and work ethics have taught me how to be a better scholar and a better leader. This work is a result of his honesty about my progress, patience and encouragement for my own explorations, and his unshakable challenge for me to be better. His professional development training has been remarkable. For his resolute, above and beyond support, I am eternally grateful to him. It is truly an honor to have had him as an advisor.

I am extremely grateful to my dissertation committee members Dr. Mahamadi Warma, Dr. Igor Griva, Dr. Carlos Rautenberg, and Dr. Zichao (Wendy) Di for strongly supporting and guiding me throughout. I thank Dr. Warma for giving me an opportunity to give a talk at the Workshop on Dynamics, Control and Numerics for Fractional PDEs in Puerto Rico. This turned out to be one of the most important conferences for me, as it brought forth a lot of positive attention to my work. I thank Dr. Di for her mentorship during my summer internships at Argonne National Laboratory, which continued into year round collaborations. My work with her has strongly influenced my research career. I am extremely grateful to her for the opportunities she has given me, and for her patience with my learning.

I would like to extend my heartfelt gratitude to Dr. Maria Emelianenko. Throughout my time at GMU, she has given me many opportunities which have all led to even greater opportunities. Her incessant support and mentorship has always shown me the light at the end of the tunnel. I am extremely grateful and honored for being nurtured by her.

I thank Dr. Matt Holzer for being the incredible teacher that he is, for being my first research mentor, and for his valuable support as I transitioned from an undergraduate to a graduate student.

I have crossed paths with some of the most wonderful colleagues like Marilyn Vasquez, Diego Torrejon, and Cigole Thomas at GMU. I extend my gratitude to them for their honest support and peer guidance. A colleague who deserves a special mention is Deepanshu Verma. He has been the most supportive peer, a wonderful collaborator, and a great friend. I am extremely grateful for his camaraderie, several constructive interactions with him, and the growth and learning that has originated from it. It is much appreciated and highly valued.

One of my greatest sources of inspiration and support has come from my family. They have been extremely patient with me and have given countless sacrifices for me to be able to pursue a career of my choice. It is an honor for me to be a pioneer, first generation Ph.D. graduate in the history of the Khatri clan. I am eternally grateful to my whole clan for their love and support.

Finally, some people who have truly enriched my journey with their incredible and impactful friendship are Sadia Hameed, Rakesh Passa, and Sanae Moummi. In addition, the support from my friends Toyam Khatri, Wei-Chieh Wang, Shab-e-Gul Rahim, Sidra

Farooq, and Asma Rubb has been a blessing. I am tremendously grateful to all these treasured friends for being there for me, their love and patience.

I also acknowledge the support of Department of Mathematical Sciences at George Mason University, NSF grants DMS-1521590, DMS-1818772, DMS-1913004, the Air Force Office of Scientific Research Award NO: FA9550-19-1-0036, the Department of Navy, Naval PostGraduate School Award NO: N00244-20-1-0005, and Provost awards at George Mason University under the Industrial Immersion Program and the Dissertation Completion Grant.

Table of Contents

List of Tables

List of Figures

Abstract

ROLE OF NONLOCAL OPERATORS IN INVERSE AND DEEP LEARNING PROBLEMS

Ratna Khatri, PhD

George Mason University, 2020

Dissertation Director: Dr. Harbir Antil

In this work, we will discuss several areas in which we harness the power of nonlocal operators. In the first part, we discuss an inverse problem from the imaging science domain. Here, we propose to use the fractional Laplacian as a regularizer to improve the reconstruction quality. In addition, inspired by residual neural networks, we develop a bilevel optimization neural network (BONNet) to learn the optimal regularization parameters, like the strength of regularization and the exponent of fractional Laplacian. As our model problem, we consider tomographic reconstruction and show an improvement in the reconstruction quality, especially for limited data, via fractional Laplacian regularization. In the second part, we propose a mathematical framework for a fractional deep neural network (fractional-DNN) for classification problems in supervised machine learning. First we formulate the deep learning problem as an ordinary differential equation (ODE) constrained optimization problem, and then we introduce a fractional time derivative based dynamical system (Neural Network) for the state equation. This architecture allows us to incorporate history (or memory) into the network by ensuring each layer is connected to the subsequent layers. The key benefits are a significant improvement to the vanishing gradient issue due to the memory effect,and better handling of nonsmooth data due to the network's ability

to approximate non-smooth functions. We test our network on several datasets for classification problems. In the third part, we introduce a new class of inverse problems for external control/source identification problems with fractional partial differential equation (PDE) as constraints. Our motivation to introduce this new class of inverse problems stems from the fact that the classical PDE models only allow the source/control to be placed on the boundary or inside the observation domain where the PDE is fulfilled. Our new approach allows us to place the source/control outside and away from the observation domain.

Chapter 1: Introduction

This work aims to introduce novel, mathematically rigorous, and computationally efficient nonlocal models for inverse problems arising in various scientific domains as well as classification problems in deep learning.

An inverse problem deals with using observed data to determine the factors that led it. In other words, it aims to find the cause from the effect. Such problems appear in numerous scientific domains like geophysics, compressed sensing, medical imaging etc. where the goal is to use the measurement data to reconstruct or recover the source of the data. These problems are often ill-posed because either the quantity of data is not enough, or it is noise-filled. This makes it quite challenging to uniquely solve these problems.

The other class of problems we have considered in this work are the classification problems which are now being increasingly handled using supervised machine learning. The goal in these types of problems is to first learn some parameters, of a prescribed model, that can classify a given set of data points. Next, the model with the learnt parameters is used to classify new data. Some examples of these problems are identification of handwritten texts, recognition and translation of spoken words, anomaly detection etc. Deep learning models are a subset of machine learning algorithms which are particularly gaining a lot of attention for such problems. The idea of deep learning is to break the learning (parameter estimation) into a number of steps. Their success, however, is mostly attributed to empirical evidence. They largely lack rigorous mathematical models which makes it challenging to fully understand and improve the learning model.

In this work, we have studied nonlocal operators stemming from Partial Differential Equations (PDE) constrained optimization problems and used them to propose improvements to the existing models of two classes of inverse problems and another class of deep learning problems.

At first, we study an inverse problem from the domain of imaging science. We consider a generalized bilevel optimization framework for solving inverse problems. To address this challenge, we introduce a novel fractional Laplacian regularization to improve the solution quality. This allows for a control on the level of smoothness in the solution. We compare our proposed regularization with the commonly used total variation regularization. We emphasize that the key advantage of using fractional Laplacian operator as a regularizer is that it leads to a linear Euler-Lagrange equation, as opposed to the total variation regularization which results in a nonlinear and possibly degenerate Euler-Lagrange equation. Next, with a regularized inverse problem model, comes the challenge of finding the optimal regularization strength. Inspired by residual neural networks, to learn the optimal strength of regularization and the exponent of fractional Laplacian, we develop a dedicated bilevel optimization neural network with a variable depth for a generic regularized inverse problem. We illustrate how to incorporate various regularizer choices into the proposed network. As an example, we consider tomographic reconstruction and show an improvement in the reconstruction quality, especially for limited data, via fractional Laplacian regularization. We successfully learn the regularization strength and the fractional exponent via our proposed bilevel optimization neural network. We show that the fractional Laplacian regularization outperforms total variation regularization. This is specially encouraging, and important, in the case of limited and noisy data. This study is briefly motivated in Section 1.1, and discussed at length in Chapter 2.

Next, we study supervised learning classification problems in deep learning. We introduce a novel algorithmic framework for a deep neural network (DNN), which in a mathematically rigorous manner, allows us to incorporate history (or memory) into the network – it ensures all layers are connected to one another. This DNN, called fractional-DNN, can be viewed as a time-discretization of a fractional in time nonlinear ordinary differential equation (ODE). The learning problem then is a minimization problem subject to that fractional ODE as constraints. We emphasize that an analogy between the existing DNN and ODEs, with standard time derivative, is well-known by now. The focus of our work is the fractional-DNN. Using the Lagrangian approach, we provide a derivation of the backward propagation and the design equations. We test our network on several datasets for classification problems. Fractional-DNN offers various advantages over the existing DNN. The key benefits are a significant improvement to the vanishing gradient issue due to the memory effect, and better handling of nonsmooth data due to the network's ability to approximate non-smooth functions. This study is briefly motivated in Section 1.2, and discussed at length in Chapter 3.

Lastly, we introduce and study a new class of inverse problems in which the control is located external to and disjoint from the observation domain. This is motivated by the article [147] which shows that for nonlocal PDEs associated with the fractional Laplacian, the classical notion of controllability from the boundary does not make sense and therefore it must be replaced by a control that is localized outside the open set where the PDE is solved. Having learned from the above mentioned result, in this work we introduce a new class of source identification and optimal control problems where the source/control is located outside the observation domain where the PDE is satisfied. The classical diffusion models lack this flexibility as they assume that the source/control is located either inside or on the boundary. This is essentially due to the locality property of the underlying operators. We use the nonlocality of the fractional operator to create a framework that now allows placing a source/control outside the observation domain. We consider the Dirichlet, Robin and Neumann source identification or optimal control problems. These problems require dealing with the nonlocal normal derivative (that we shall call interaction operator). We create a functional analytic framework and show well-posedness and derive the first order optimality conditions for these problems. We introduce a new approach to approximate, with convergence rate, the Dirichlet problem with nonzero exterior condition. The numerical examples confirm our theoretical findings and illustrate the practicality of our approach. This study is briefly motivated in Section 1.3, and discussed at length in Chapter 4.

In the sections below, we give a brief formal introduction to the three models discussed above. We remark that the notations in the following sections and their corresponding chapters are self-contained.

1.1 Fractional Laplacian Regularization and Parameter Estimation

Inverse problems such as

$$\min_u J(u), \quad \text{with} \quad J(u) := \frac{1}{2}\|Ku - f\|^2,$$

where K is the forward map, f is the measurement data, and u is the sample feature (e.g. pixel intensity of an image) that we need to recover, are typically ill-posed. This is due to limited data and imperfection of experiments (thus noisy data). One process of making this problem better posed is to enforce some prior knowledge of the solution (e.g. smoothness or sparsity features) in the form of a regularization, which is an addition of a term to the objective functional $J(u)$. The challenges associated with solving a regularized inverse problem, then, are the choice of regularization and the regularization strength.

Recently, in [8] we have introduced a new form of regularization for inverse problems, i.e., fractional Laplacian $(-\Delta)^s$ with $0 < s < 1$, see [6]. The regularized problem is

$$\min_u \ J(u) + R(u, \mu),$$

with

$$\mathcal{R}(u, \mu) := \frac{1}{2}\|\sqrt{\lambda}(-\Delta)^{\frac{s}{2}}u\|^2,$$

where $\mu = (\lambda, s)$. The strength of regularization is given by λ and the order of smoothness is dictated by s.

We remark that such a regularization enforces a reduced smoothness than, for instance, H^1-regularization, and yields linear equation as the optimality condition. The latter is in contrast to the state-of-the-art regularization such as total-variation which leads to nonlinear/degenerate optimality condition [122]. Our numerical results for fractional Laplacian in [8] illustrate far better results than TV, see [6] for a mathematical justification.

To find μ, we have created a novel Bilevel Optimization Neural Network which shares some similarity with a deep Residual Neural Network (RNN) [83], but is mathematically rigorous. BONNet has a supervised learning (*training*) phase in which it uses existing data to learn μ. This μ is then used in the inference (*testing*) phase to reconstruct/recover solution of the inverse problem for new (previously unseen) data. An added feature of BONNet is that, within an optimization framework, it easily allows imposing inequality constrains [139] which is difficult for a standard RNN [107].

1.2 Deep Neural Networks with Memory

Recently, in machine learning DNNs have gained a lot of attention due to their superior performance in various domains like healthcare, autonomous vehicles etc. DNNs are machine learning architectures whose overarching goal is to approximate $\mathcal{F}$, an input (Y_0) to output (Y_N) map:

$$Y_j = \mathcal{F}(Y_{j-1}), \quad j = 1, \cdots, N; \;\; N > 1.$$

In the case of supervised learning problems, the model is approximated using data with a known classification. The approximated model is then used to make classification prediction of new data points. A popular example of a DNN from imaging science is the ResNet [83] which is an RNN. An RNN helps approximate the map $\mathcal{F}$ by inducing an identity map between the layers.

The articles [68, 76] show that an RNN is equivalent to a time-discretization of a continuous ODE. Thus in the continuous setting, the learning problem becomes minimizing the loss function $\mathcal{J}$ at the final time T subject to the ODE constraints:

$$\begin{aligned} &\min_{\theta=(K,b)} \quad \mathcal{J}(\theta, (Y(T), C)) \\ \text{s.t.} \quad & d_t Y(t) = \sigma(K(t)Y(t) + b(t)), \quad t \in (0, T), \\ & Y(0) = Y_0. \end{aligned} \tag{1.2.1}$$

where (Y_0, C) pair is the training data and (K_i, b_i) pair denotes the unknown weight and bias. DNNs, including RNNs, suffer from various challenges such as vanishing gradients [30, 142], inability to approximate non-smooth functions, long training time, etc. A dense neural network (DenseNet) has been introduced to overcome the first issue, however, it is only an adhoc method [88]. Rigorous approaches to approximate non-smooth functions, e.g., absolute value $|x|$ are scarce [89].

In this work, we shall consider a *fractional-in-time* derivative based ODE constraint for this setup. We propose to use Caputo fractional time derivative [12],

$$d_t^\gamma u = \frac{1}{\Gamma(1-\gamma)} \frac{d}{dt} \int_0^t \frac{u(r) - u(0)}{(t-r)^\gamma} dr.$$

The resulting neural network will be called *fractional-DNN*; whose key advantage is the fact that fractional time derivative allows memory, which in turn allows connectivity of all network layers. This in turn helps overcome the vanishing gradient issue and it can be applied to non-smooth functions such as absolute value. In our numerical experiments, we consider various datasets and show the performance of our proposed network.

1.3 External Optimal Control

There are many real life applications which require that the control element be placed outside and away from the effective domain. This is in contrast to what the typical classical models permit. Consider the classical diffusion models which only allow the control or source placement either inside the effective domain Ω or on its boundary $\partial\Omega$. This can be prohibitive for high consequence applications such as magnetic drug targeting where the control (magnets) are away from the boundary [13]. Motivated by this fact, in [10], we have introduced a new class of optimal control problems that allows control/source placement outside and away from the boundary. We achieve this via the nonlocality of fractional Laplacian operator, $(-\Delta)^s$ with $s \in (0,1)$. Recall that in Section 1.1, the fractional Laplacian was used as a regularizer which allows control on the smoothness. In this case, the full force of nonlocality of this operator is used to facilitate exterior control in an optimal control problem subject to a fractional elliptic equation as constraint.

For illustration, let us consider a simplified geometry in Figure 1.1, in which we have a source/control (*small square*) located outside the observation domain Ω (*large square*) where the fractional PDE is satisfied. The class of inverse problems that we have introduced is a

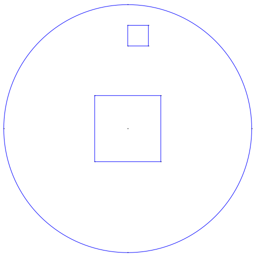

Figure 1.1: Schematic of Ω where PDE is fulfilled (large square) and control domain (small square).

minimization problem subject to a fractional elliptic equation, given by

$$\min_{z \in Z_{ad}} \ \frac{1}{2}\|u - u_d\|^2_{L^2(\Omega)} + \frac{\xi}{2}\|z\|^2_{L^2(\mathbb{R}^N \setminus \Omega)}$$

$$\text{s.t.} \quad \begin{cases} (-\Delta)^s u = 0 & \text{in } \Omega \\ \\ u = z & \text{on } \mathbb{R}^N \setminus \Omega, \end{cases}$$

Here, u_d is the data from the observation domain. The state u satisfies the fractional Poisson equation inside the observation domain, and it is equal to the control z outside the observation domain. Then, given data/measurements $u_d \in L^2(\Omega)$ and $0 < s < 1$, our goal is to invert for the source z located outside Ω.

For this problem, we study the notions of weak and very weak solutions and show how to approximate them. We discuss several examples, including the schematic shown in Section 1.3. In particular, we establish that for small values of the fractional exponent s, we are better able to achieve our aim of locating the external source.

Chapter 2: Bilevel Optimization, Deep Learning and Fractional Laplacian Regularization with Applications in Tomography [1]

2.1 Introduction

Inverse problems appear in numerous scientific domains, such as medicine, geophysics, astronomy, computer vision, and imaging etc. However, they are typically ill-posed, due to the limited data and imperfection of experiments, and require some form of regularization [66, 77, 87, 99, 113]. Two key challenges are associated with solving a regularized inverse problem. The first is the choice of regularization. Among the most popular choices, the total variation regularization [122, 130] is of edge-preserving nature. However, its non-differentiability makes its usage numerically challenging. Another choice is the Tikhonov regularization [138], which has a smoothing property. Each choice, however, comes with its own challenges such as nonlinearity, non-smoothness, over-smoothing etc. The second associated challenge is to choose the strength of the regularization, usually dictated by the parameter μ, for which there is no consensus.

Recently, deep learning approaches such as Convolution Neural Networks (CNN) and Residual Neural Networks have shown remarkable potential in image classification and reconstruction where, often, the goal is to learn the whole regularizer [78, 153, 154]. These approaches, however, may not be robust in general [106, 109]. Firstly, learning problems are usually nonconvex, and the local minima may be sensitive to the initialization of parameters and the choice of optimization method. Secondly, these approaches often do not incorporate the domain-specific knowledge of the system (e.g., the known solution features) directly

[1] This work has been published in [8].

into the network, for instance. In addition, they often lack a mathematical justification [61, 67, 117, 124, 150]. The main contributions of this work are two-folds:

(a) Extend the fractional Laplacian introduced in [6] as a regularizer to the general setting of a linear inverse problem.

(b) Instead of learning the entire regularizer, we consider a bilevel optimization scheme to learn the strength of the regularization and the fractional exponent based on the prior knowledge of the system. More specifically, we set up a *bilevel optimization neural network (BONNet)*. In this network, the upper level objective measures an expectation of the reconstruction error over the training data while the lower level problem measures the regularized data misfit.

There are several existing attempts to take advantage of machine learning to improve the solution quality. The most common way is to explore neural network as a post-processing step to refine the solution obtained by base-line methods (e.g., iterative method or filtered back projection [92]), see also [91, 129].

Our approach is closely related to the methodology introduced in [78]. In fact, ours can be thought as a special case in the case of total variation regularization, where the authors consider a variational model for reconstruction of MRI data. The authors focus on a generalized total variation model (Fields of Experts model) and also learn the underlying parameters. For completeness we also refer to [45] for a discussion on bilevel optimization. We emphasize that the main novelty in our work is the use of fractional Laplacian [6, 42, 135] as a regularizer and learning the fractional exponent with an application to tomographic reconstruction. The fractional Laplacian introduces nonlocality and tunable regularity. Another type of parameter search strategy has been proposed in [50] where the authors consider Tikhonov-based regularizations, and propose a machine learning based strategy to learn the strength of regularization. Their scheme is based on the generalized singular value decomposition (GSVD), or its approximation, of the forward operator and the regularization operator pair. However, computing GSVD can be computationally challenging [79]. Our approach differs from the existing works as we propose to use the fractional Laplacian as a regularizer, which is cheaper to evaluate, and allows us to enforce the prior knowledge of the sample features, including smoothness and sparsity. The fractional Laplacian has been successfully applied in image denoising [6, 18], geophysics [149], diffusion maps [7], biology [41], novel exterior optimal control [10, 19], etc. We also emphasize that our proposed framework is flexible, for it can easily incorporate inequality constraints (on the optimization variables), which can be solved by a large number of existing solvers, and directly generalizes to other types of regularizations such as the p-Laplacian [38, 102]. Therefore, our proposed framework brings machine learning closer to the traditional optimization. Notice that the machine learning algorithms are still in their infancy when it comes to handling constraints, see, for instance, [107] and the references therein.

The numerical examples presented in this work are strongly motivated by tomographic reconstruction, see Subsection 2.2.3. Further realistic application of interest to us is the MRI reconstruction, considered in [78]. It is also of interest to implement our approach in open source Python packages such as TensorFlow and PyTorch. These would be considered as a part of the future work.

The rest of the Chapter is organized as follows. In Section 2.2, we introduce the mathematical formulation of the standard linear inverse problem with regularizers. In particular, we consider the fractional Laplacian as a regularizer for inverse problems. We show a comparison of fractional Laplacian and total variation as regularizers for a tomographic reconstruction problem. Section 2.3 is devoted to our proposed algorithmic framework, BONNet, to learn the optimal regularization strength, as well as the order of the fractional Laplacian. In Section 2.4, we provide further numerical experiments illustrating the application of BONNet to the tomographic reconstruction problem.

2.2 Regularization in Inverse Problems

The regression model for data misfit in inverse problems is given by

$$\min_{u} J(u) := \frac{1}{2}\|Ku - f\|^2_{L^2(\Omega)}, \tag{2.2.1}$$

where $f : \Omega \mapsto \mathbb{R}$ is a given function and $\Omega \subset \mathbb{R}^n$ with $n \geq 1$ is a bounded domain. Here K is the forward map, which we assume is a bounded linear operator on $L^2(\Omega)$ where the latter denotes the square integrable functions. Moreover, u is the sample feature that we want to recover, or reconstruct. The ill-posed nature of (2.2.1) makes it almost necessary to consider regularization in the wake of often noise-filled data; owing to the imperfections in the data gathering process. Therefore, we consider a regularized regression model to improve the solution quality. In a more general sense, let $\Omega \subset \mathbb{R}^n$ with $n \geq 1$ be a bounded Lipschitz domain with boundary $\partial\Omega$, $f : \Omega \to \mathbb{R}$ be an $L^2(\Omega)$ function (given datum), $K : L^2(\Omega) \to L^2(\Omega)$ be a bounded linear operator, and X be a Banach space. Then a standard regularized variational model is given by

$$\min_{u \in X_{ad} \subseteq X} J(u) := \frac{1}{2}\|Ku - f\|^2_{L^2(\Omega)} + \mathcal{R}(u, \mu), \tag{2.2.2}$$

where X_{ad} is a closed, convex, nonempty admissible set which is contained in the solution space X, and u is the solution that we want to reconstruct or recover. Some examples of the operator K for inverse problems in imaging science are the identity operator (image denoising problem) [122], convolution operator (image deblurring problem) [85,86], and the Fourier or wavelet transforms [134]. Therefore, in (2.2.2), the first term prevents the forward simulation from departing "too far" away from f, thus it helps maintain the fidelity to f. In the absence of the second term ($\mathcal{R}(u, \mu)$), (2.2.2) may be ill-posed [80]. The regularizer $\mathcal{R}(u, \mu)$ incorporates prior knowledge of the sample (like smoothness, sparsity, etc.), where μ balances the data misfit and the penalty enforced by the regularizer. Various choices of

$\mathcal{R}(u,\mu)$ have been proposed in the literature. In this work, we focus on the tomographic reconstruction problem, regularized with the fractional Laplacian, and compare it against the total variation regularization.

2.2.1 Total Variation Regularization

The penalty term for total variation (TV) regularization is given by

$$\mathcal{R}(u,\mu) = \lambda\, \mathrm{TV}(u), \tag{2.2.3}$$

where $\mu = \lambda$ is a scalar. Here, $\mathrm{TV}(u)$ denotes the total variation semi-norm on Ω and $X = BV(\Omega) \cap L^2(\Omega)$, where $BV(\Omega)$ denotes the set of functions of bounded variations [4]. Formally speaking, $\mathrm{TV}(u) := \int_\Omega |\nabla u|$ and as a result the corresponding Euler-Lagrange equation for (2.2.2) is: Find $u \in X_{ad} \subset X$ such that

$$\left\langle -\lambda \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + K^*(Ku - f), \hat{u} - u \right\rangle_{X',X} \geq 0, \quad \forall\ \hat{u} \in X_{ad} \tag{2.2.4}$$

i.e., a nonlinear and possibly degenerate (due to $1/|\nabla u|$) variational inequality which is challenging to solve. We remark that X' is the dual of X and K^* is the adjoint of K. Designing solvers for (2.2.4) is still an active area of research [27]. The success of $\mathrm{TV}(u)$ can be attributed to the fact that it prefers to fit shorter curves over the longer ones, thus avoids fitting noise and enforces sparsity. Additionally, it enforces much weaker regularity than the H^1-regularization, i.e., when $\mathcal{R}(u,\mu) = \frac{\lambda}{2}\int_\Omega |\nabla u|^2$, with $\mu = \lambda$, and as a result it is possible to capture desirable sharp transitions in the reconstruction [122].

2.2.2 Fractional Laplacian Regularization

The fractional Laplacian as a regularization for (2.2.2) is given by,

$$\mathcal{R}(u,\mu) = \frac{1}{2}\|\sqrt{\lambda}(-\Delta)^{\frac{s}{2}}u\|^2_{L^2(\Omega)}, \tag{2.2.5}$$

where $\mu = (\lambda, s)$ is a vector. Moreover, with $0 < s < 1$, and $(-\Delta)^s$ denoting the fractional power of the classical Laplacian defined, for instance, in a spectral sense [6, 135]. We remark that such a regularization enforces a reduced smoothness than H^1-regularization. The extent of the smoothness is dictated by the fractional power 's'. The key advantage of using this regularization is that the resulting Euler-Lagrange equation for (2.2.5) is: Find $u \in X_{ad}$

$$\big\langle \lambda\,(-\Delta)^s u + K^*(Ku - f), \hat{u} - u \big\rangle \geq 0, \qquad \forall\, \hat{u} \in X_{ad} \tag{2.2.6}$$

i.e., a variational equation with a linear operator. Such a problem has a unique solution in the fractional order Sobolev space $X = H^s(\Omega)$ [95]. This regularization has been applied successfully in image denoising [6] (with $K = I$, but with $u \in X$, instead of X_{ad}, as a result the variational inequality (2.2.6) becomes an equality).

2.2.3 Tomographic Reconstruction

Tomographic reconstruction is a noninvasive imaging technique with the goal of recovering the internal characteristic of a 3D object using a penetrating wave. It has shown revolutionary impact on various fields including physics, chemistry, biology, and astronomy. In a tomographic scan, a beam of light (e.g., X-ray) is projected onto the object to generate a 2D representation of the internal information along the beam path. By rotating the object, a series of such 2D projections are collected from different angles of view, collectively known as a sinogram (measurement data f), which can then be used to recover the internal characteristics (e.g., the attenuation coefficient) of the object [55] (see Figure 2.1). However, the limited data, due to the discrete nature of the physical experiment and dosage limits,

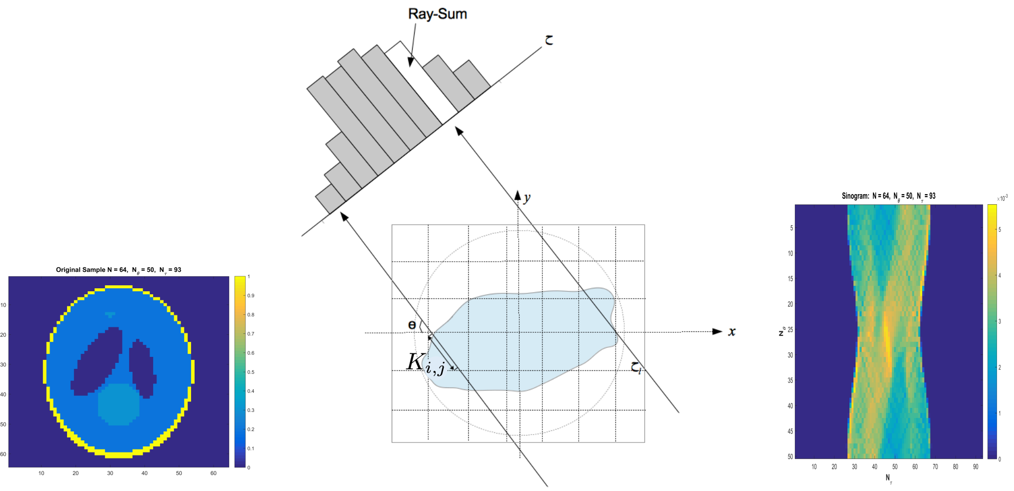

Figure 2.1: Geometric sketch of X-ray tomography (*middle*) which maps the sample (*left*) from the (x, y) space to the sinogram (*right*) on the (τ, θ) space.

makes the reconstruction problem ill-posed, i.e., many local minima exist for the objective function which is used to describe the discrepancy between the forward model and the measurement data. For illustration purpose, we confine ourselves to reconstruct 2D objects. The mathematical foundation of tomography is the Radon transform [118], for which K is defined as,

$$Ku(\tau, \theta) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, y)\delta(\tau - x\cos\theta - y\sin\theta)\, dx\, dy, \tag{2.2.7}$$

where $u : \mathbb{R}^2 \mapsto \mathbb{R}$ is compactly supported on a bounded domain $\Omega \subset \mathbb{R}^2$ and δ is the Dirac mass, $\tau \in [0, \infty)$ and $\theta \in [0, 2\pi)$ define the line of the beam path in a restricted domain. In practice, we can not recover the object at all points in space. Instead, we discretize Ω as $N \times N$ uniform pixels. Given N_θ number of angles and N_τ number of discrete beamlets, our goal is to recover the piecewise constant approximation (on each pixel) $u \in \mathbb{R}^{N^2}$. Correspondingly, the discrete form of operator K is the matrix $\mathbf{K} = (k_{i,j})_{i,j=1}^{N_\theta N_\tau, N^2}$ where the entries $k_{i,j}$ denote the contribution of jth pixel of u to the ith component of the generated data.

2.2.4 Comparison of Fractional Laplacian with TV for Tomographic Reconstruction

To show the benefit of fractional Laplacian, we compare its performance against TV regularizer on a model problem. For now, we use a well-known, but not necessarily efficient, criterion to choose λ and a fixed fractional exponent 's' for this preliminary comparison. The rigorous computation of optimal (λ, s) will be part of a forthcoming discussion.

We choose our test problem as the tomographic reconstruction. First we synthetically generate the tomographic measurements of the sample u by taking its discrete Radon transform, which gives us the data f. The sample u and its corresponding sinogram f are illustrated in Figure 2.1. To get the noisy data, we add 0.1% Gaussian noise to f. More details on tomographic reconstruction is provided in Section 2.4. Next we show the reconstructions based on the two regularizers, namely the fractional Laplacian (2.2.5) and the total variation (2.2.3), in Figure 2.2. The *left* panel corresponds to reconstructions based on sinogram f without noise, and the *right* panel corresponds to reconstructions based on noisy f. *Rows 1* and *2* pertain to total variation and fractional Laplacian regularization, respectively.

In the absence of noise, the reconstructions based on both regularizers are comparable. However, noiseless data does not depict a realistic situation [53]. In reality, the actual experimental data is always noisy due to the imperfections in the data acquisition process. We note that for noisy data, particularly for the fewer projection case with $N_\theta = 10$ angles, fractional Laplacian regularization gives better reconstructions than the total variation regularization. This can be specifically seen in Figure 2.2 (*right panel, row 2*) where finer features are better recovered e.g. the small circle at the bottom. However, to fully explore the potential of regularization technique, the well-known challenge is to find the appropriate regularization strength λ to optimally balance the trade-off between data misfit and prior knowledge enforcement. In the case of fractional Laplacian regularization, the exponent 's'

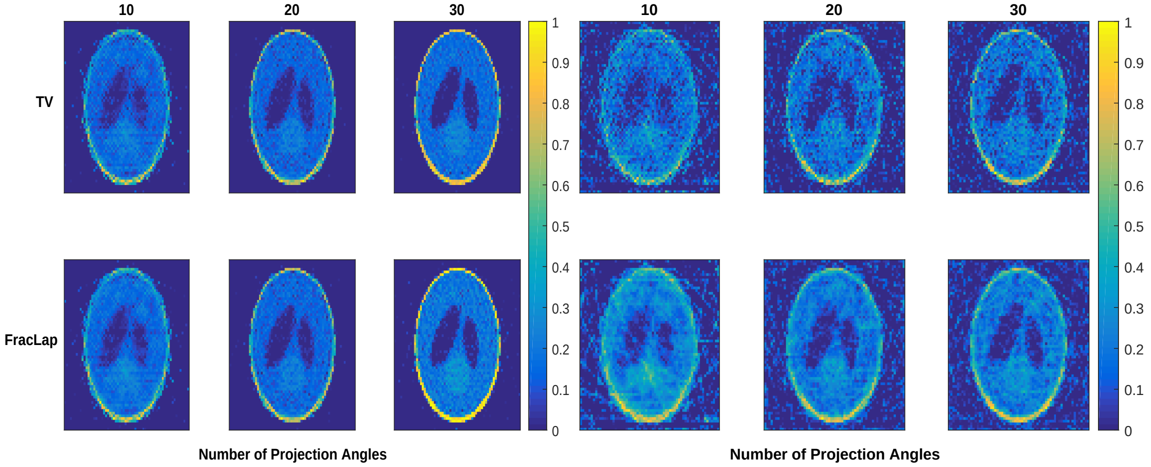

Figure 2.2: Tomographic reconstructions based on the total variation regularization (*row 1*) and fractional Laplacian (with $s = 0.4$, *row 2*) for data without noise (*left*) and with 0.1% noise (*right*). The fractional Laplacian outperforms the total variation regularization in recovering finer features as well as in retaining high intensity regions, specially when the data is noisy and highly under-sampled.

only complicates the parameter choice further.

For the reconstructions in Figure 2.2, given a wide range of values for $\lambda \in [1 \times 10^{-18}, 10]$, we fix $s = 0.4$ (motivated by the first author's prior experience in [6]), and solve the minimization problem (2.2.2) using an inexact truncated-Newton method for bound-constrained problems [111]. The optimal value of λ is then chosen using a combination of L-curve criterion [81] and the lowest ℓ_2-norm of the reconstruction error compared to the ground of truth. When L-curve criterion fails, we solely rely on the lowest ℓ_2-norm. In our experience, this behavior is true for both TV and fractional Laplacian. As a result, the optimal values of λ for these tests is found to be in the range $[1 \times 10^{-10}, 1]$. This procedure of finding an optimal λ is labor-intensive, and requires access to the true solution, which is not available in practice. We remark that, to our experience, L-curve is efficient (not necessarily optimal) only in the case of strongly convex regularization which is definitely not the case with fractional Laplacian when 's' is also considered as a regularization parameter (non-convex with respect to 's'). L-curve criterion requires many different trial values of λ, along with a good guess of the interval to locate the corner of the L-curve. This requires a lot of human-intervention and fine-tuning. Furthermore, the regularized solution obtained by the λ predicted by L-curve sometimes fails to converge to the true solution [144].

The next section addresses the issue of finding the optimal regularization parameters by proposing a deep bilevel optimization neural network.

2.3 Parameter Learning via Bilevel Optimization Neural Network

Parameter search lies at the core of optimization. In particular, we seek parameters corresponding to the strength of regularization, which is a persistent challenge in the scientific community. To this end, we introduce a learning based approach as adverted in Section 2.1. We first state a generic bilevel optimization problem,

$$\begin{aligned} &\min_{\mu \in \mathcal{M}_{ad}} \quad \phi(\mu) \\ \text{s.t.} \quad &\min_{u \in X_{ad}} J(u, \mu) := \frac{1}{2}\|Ku - f\|^2_{L^2(\Omega)} + \mathcal{R}(u, \mu), \end{aligned} \tag{2.3.1}$$

where $\mathcal{M}_{ad}$ is a closed convex and nonempty admissible set for μ.

In Subsection 2.3.1, motivated by [78], we present a machine-learning based approach to learn the regularization strength for a generic choice of regularizer. One of the key novelty of this work is to use fractional Laplacian as the regularizer. Notice that the lower level problem (2.2.2) in (2.3.1) can be solved using the existing techniques.

2.3.1 Bilevel Optimization Neural Network (BONNet)

Recently, deep residual learning has received a tremendous amount of attention in machine learning for its immense potential to overcome the challenges faced by the traditional deep learning architectures, such as training complexity and vanishing gradients. These are resolved by adding skip connections, which transfer information between the layers [83]. Deep residual learning has enabled remarkable progress in imaging science [83, 91, 152], biomedical applications [48, 78, 100], satellite imagery, remote sensing [35, 137, 155], etc. In our work, we use the potential of deep learning to learn the regularization parameter μ which, for instance, contains the strength λ and the fractional exponent 's'. We propose a dedicated deep bilevel optimization neural network to learn the regularization parameters. Our goal is to solve (2.3.1) for which we seek our modeling inspiration from [78], and define $\phi(\mu)$ as the average mean squared error over m distinct samples, i.e.,

$$\phi(\mu) := \frac{1}{2m} \sum_{i=1}^{m} \|u^{(i)}(\mu) - u_{true}^{(i)}\|_{L^2(\Omega)}^2,$$

where $u(\mu)$ solves the lower level problem in (2.3.1), and corresponds to the sample characteristic that we wish to recover or reconstruct. Moreover, u_{true}, as the name suggests, is the *known* true solution.

We emphasize a few novelties of this work: first, our proposed network works directly on the data space, as opposed to the image space as a post-processing step as in [91, 129]. Second, it generalizes to any bounded linear operator K (the forward map; which defines the physics of the underlying system) and any $\mathcal{R}(u, \mu)$ (the regularization term; which allows us to incorporate the domain-specific knowledge of the solution). Third, we propose the use of fractional Laplacian as a regularizer with tunable regularity/smoothness. We also show how to integrate this choice of regularization into the BONNet architecture. We remark that fractional Laplacian introduces nonlocality in BONNet, which is challenging from both analytical and computational point of view.

We first define the notion of a generalized regularizer and the projection map that we will be using to define the BONNet architecture.

- **Generalized Regularizer.** Let $u(\mu)$ be the solution of the inner problem in (2.3.1) which depends on μ. Notice that the inner problem in (2.3.1) is same as (2.2.2). Let $T := T(\mu, u(\mu))$ be the action of some linear or nonlinear operator acting on $u(\mu)$, and $\sigma := \sigma(T)$ be a function. Then, we define a **generalized regularizer** as,

$$\mathcal{R}(u, \mu) := \mathcal{R}(\sigma(T)) = \frac{1}{2}\|\sigma(T(\mu, u(\mu)))\|^2_{L^2(\Omega)}. \tag{2.3.2}$$

Then, for m distinct samples, we can write our inner minimization problem (2.2.2) with a generalized regularizer as an average over m samples, and $\mu \in \mathcal{M}_{ad}$,

$$\min_{u\in X_{ad}} J(u, \mu) := \frac{1}{2m}\sum_{i=1}^{m} \left[\|Ku^{(i)} - f^{(i)}\|^2_{L^2(\Omega)} \; + \; \|\sigma\|^2_{L^2(\Omega)}\right]. \tag{2.3.3}$$

To solve this inverse problem, we will employ derivative based methods such as projected gradient descent. The directional derivative of J in a direction h in (2.3.3) w.r.t. u in its variational form is; for each sample, $i = 1, ..., m$,

$$\begin{aligned} DJ(u^{(i)}, \mu)[h] = \frac{1}{m}\Big[&(K^*(Ku^{(i)} - f^{(i)}), h)_{L^2(\Omega)} + \\ &\left((\partial_{u^{(i)}}T)^* \, (\partial_T\sigma)\sigma, h)_{L^2(\Omega)}\Big]. \right. \end{aligned} \tag{2.3.4}$$

- **Solver: Projected Gradient Descent Method** The choices of X_{ad} and $\mathcal{M}_{ad}$ are problem dependent, for example, for tomographic reconstruction model, we let $X_{ad} := \{u \in X \mid u \geq 0\}$. Moreover, we set $\mathcal{M}_{ad} := \Lambda_{ad}$ for total variation and $\mathcal{M}_{ad} := \Lambda_{ad} \times S_{ad}$ where $\Lambda_{ad} := \{\lambda \in \mathbb{R} \mid \lambda \geq \epsilon_1 > 0\}$ and $S_{ad} := \{s \in \mathbb{R} \mid 0 < \epsilon_2 \leq s \leq 1-\epsilon_2\}$ for the fractional Laplacian. See Subsection 2.4.1.2 for more details on this application. In order to satisfy these constraints, we use the *projected gradient descent method with line search* [93] to solve our inner and outer minimization problems in (2.3.1). Then, the projected gradient descent scheme for solving (2.3.3), for a fixed μ, n iterations (*depth of the network*), α as the line search parameter (i.e. *the learning* *rate*), u_0 as the initial guess, for the network layers (optimization iteration) $j = 1, ..., n$, is given by

$$u_j^{(i)} = P_{X_{ad}} \left(u_{j-1}^{(i)} - \alpha \nabla_{u_{j-1}^{(i)}} J(u_{j-1}^{(i)}, \mu) \right). \tag{2.3.5}$$

where $P_{X_{ad}}(\cdot)$ denotes the projection on the admissible set X_{ad}, see Subsection 2.4.1.2 for more details on the tomographic reconstruction application. Note that, (2.3.5) is also known as the *forward propagation*. We are using ∇ to denote the gradient and D to denote the directional derivative (cf. (2.3.4)). Now substitute the gradient from (2.3.4) in (2.3.5) to arrive at,

$$u_j^{(i)} = P_{X_{ad}}\Big(u_{j-1}^{(i)} - \frac{\alpha}{m}\big[K^*(Ku_{j-1}^{(i)} - f^{(i)}) + (\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma\big]\Big). \tag{2.3.6}$$

To compute the learning rate α, we use line search for projected gradient descent as described in [93, pg. 91].

Putting it all together, we now describe our proposed **BONNet** architecture. Suppose we have m distinct samples, and n layers in our network. Let $u_{true}^{(i)}$ and $f^{(i)}$ be the known true solution and its corresponding experimental data for the *ith* sample, with $i = 1, ..., m$. Then, we formulate our bilevel supervised learning problem as; for $j = 1, ..., n$,

$$\begin{aligned} &\min_{\mu \in \mathcal{M}_{ad}} \phi(\mu) = \frac{1}{2m} \sum_{i=1}^{m} \|u_n^{(i)}(\mu) - u_{true}^{(i)}\|_{L^2(\Omega)}^2 \\ &\text{s.t. } u_j^{(i)} = P_{X_{ad}} \left(u_{j-1}^{(i)} - \frac{\alpha}{m}[K^*(Ku_{j-1}^{(i)} - f^{(i)}) + (\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma] \right). \end{aligned} \tag{2.3.7}$$

Remark 2.3.1 (**Relation to Existing Neural Networks**). Notice the resemblance between the inner level problem in (2.3.7) and a residual neural network [37, 83], see also for other related works [71, 82, 84]. Indeed, after rewriting we obtain that

$$u_j^{(i)} = P_{X_{ad}} \left(\mathcal{L} u_{j-1}^{(i)} + b - \frac{\alpha}{m} (\partial_{u_{j-1}^{(i)}} T)^* (\partial_T \sigma) \sigma \right)$$

where $\mathcal{L} := \left(I - \frac{\alpha}{m} K^* K\right)$, $b := \frac{\alpha}{m} K^* f^{(i)}$. The first two terms $\mathcal{L} u_{j-1}^{(i)}$ and b are available in a typical neural network. The last term $-\frac{\alpha}{m}(\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)$, which is not always affine in $u_{j-1}^{(i)}$, can be thought as an action of an activation function. We further emphasize that the projection $P_{X_{ad}}$ is another ReLU type activation function.

To solve the outer level problem for $\mu \in \mathcal{M}_{ad}$ we again use the projected gradient descent method, as described above, with learning rate β and q iterations,

$$\mu_{l+1} = P_{\mathcal{M}_{ad}} \left(\mu_l - \beta \, \nabla_{\mu_l} \phi(\mu_l) \right), \qquad l = 0, ..., q-1, \tag{2.3.8}$$

where $P_{\mathcal{M}_{ad}}(\cdot)$ is the projection onto the admissible set. It then remains to evaluate $\nabla_{\mu_l} \phi(\mu_l)$. For the remainder of the discussion, we shall assume that $u_n^{(i)}$ is sufficiently smooth with respect to μ. After applying the chain rule, we obtain that

$$\nabla_{\mu_l} \phi(\mu_l) = \frac{1}{m} \sum_{i=1}^{m} \int_\Omega (u_n^{(i)} - u_{true}^{(i)}) \left. \frac{d u_n^{(i)}}{d\mu} \right|_{\mu=\mu_l} d\Omega. \tag{2.3.9}$$

As noted earlier, the most challenging part of this network is the computation of sensitivity of u w.r.t. μ, because at each network layer, u depends on the previous iterate, as well as μ, as can be seen in the lower level problem in (2.3.7). We evaluate $\left. \frac{du_n^{(i)}}{d\mu} \right|_{\mu=\mu_l}$ in (2.3.9) by implicit differentiation. This results in an iterative system of equations that we need to solve. For each sample index 'i', it is explicitly derived as follows, for $j = 1, ..., n$

$$\left.\frac{du_j}{d\mu}\right|_{\mu=\mu_l} = \frac{\partial u_j}{\partial u_{j-1}} \cdot \left.\frac{du_{j-1}}{d\mu}\right|_{\mu=\mu_l} + \frac{\partial u_j}{\partial \mu} \cdot \left.\frac{d\mu}{d\mu}\right|_{\mu=\mu_l}, \tag{2.3.10}$$

where,

$$\begin{aligned}\frac{\partial u_j}{\partial u_{j-1}} = I - \frac{\alpha}{m}\Bigg[& K^*K + \frac{\partial}{\partial u_{j-1}}\Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\Big(\frac{\partial \sigma}{\partial T}\Big)^*\sigma + \\ & \Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\frac{\partial}{\partial u_{j-1}}\Big(\frac{\partial \sigma}{\partial T}\Big)^*\sigma + \\ & \Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\Big(\frac{\partial \sigma}{\partial T}\Big)^*\Big(\frac{\partial \sigma}{\partial T}\cdot\frac{\partial T}{\partial u_{j-1}}\Big)\Bigg],\end{aligned} \tag{2.3.11}$$

and,

$$\begin{aligned}\frac{\partial u_j}{\partial \mu} = -\frac{\alpha}{m}\Bigg[& \Big(\frac{\partial}{\partial \mu}\Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\Big)\Big(\frac{\partial \sigma}{\partial T}\Big)^*\sigma + \\ & \Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\Big(\frac{\partial}{\partial \mu}\Big(\frac{\partial \sigma}{\partial T}\Big)^*\Big)\cdot\sigma + \\ & \Big(\frac{\partial T}{\partial u_{j-1}}\Big)^*\Big(\frac{\partial \sigma}{\partial T}\Big)^*\cdot\frac{\partial \sigma}{\partial T}\frac{\partial T}{\partial \mu}\Bigg].\end{aligned} \tag{2.3.12}$$

Substituting (2.3.11) and (2.3.12) in (2.3.10) yields the sensitivity of u w.r.t. μ. Now that we have the key architecture of the deep BONNet, we divide our network into a *training* phase and a *testing* phase, as is common in a standard machine learning framework. During the *training* phase, we solve the bilevel optimization problem (2.3.7) to learn the regularization parameters, and during the *testing* phase we only solve the inner problem in (2.3.7) using the regularization parameters learned from the training phase. The training phase can be carried out offline (i.e. in advance), and testing phase can be carried out online (i.e. as the experimental data becomes available).

2.3.1.1 General Framework of BONNet

We summarize the training and testing phases of our deep BONNet architecture as follows:

- ***Training Phase*** **(Algorithm 1).** In this phase, we pass in m training samples $\left\{u_{true}^{(i)}, f^{(i)}\right\}_{i=1}^{m}$ to learn the *optimal* μ which we denote by μ^*. The depth of the deep BONNet at the *training* phase is 'q sets of n layers'. This phase can be carried out offline.

Algorithm 1 Training Phase of BONNet

```
Input: $\left\{u_{true}^{(i)}, f^{(i)}\right\}_{i=1}^{m}$, $m$ training samples
Output: $\mu^*$
Initialize $u_0$, $\frac{du_0}{d\mu}$ and $\mu_0$
for for $l = 0$ to $q-1$ do
    for for $j = 1$ to $n$ do
        Compute $u^{(i)}$ and $\frac{du_n^{(i)}}{d\mu}$ for all $i = 1, ..., m$:
            $u_j^{(i)} = P_{X_{ad}}\Big(u_{j-1}^{(i)} - \frac{\alpha}{m}\left[K^*(Ku_{j-1}^{(i)} - f^{(i)}) + (\partial_{u_{j-1}^{(i)}}T)^*(\partial_T\sigma)\sigma\right]\Big).$
        {Compute $\alpha$ using line search as discussed in Subsection 2.3.1}
            $\left.\frac{du_j^{(i)}}{d\mu}\right|_{\mu=\mu_l} = \frac{\partial u_j^{(i)}}{\partial u_{j-1}^{(i)}} \cdot \left.\frac{du_{j-1}^{(i)}}{d\mu}\right|_{\mu=\mu_l} + \frac{\partial u_j^{(i)}}{\partial \mu} \cdot \left.\frac{d\mu}{d\mu}\right|_{\mu=\mu_l}$
        {See (2.3.11), (2.3.12) for explicit expressions}
    end for
    Compute the gradient of $\phi(\mu)$:
        $\nabla_{\mu_l}\phi(\mu_l) = \frac{1}{m}\sum_{i=1}^{m}\int_\Omega (u_n^{(i)} - u_{true}^{(i)}) \left.\frac{du_n^{(i)}}{d\mu}\right|_{\mu=\mu_l} d\Omega,$
    Update $\mu$:
        $\mu_{l+1} = P_{\mathcal{M}_{ad}}\left(\mu_l - \beta\nabla_{\mu_l}\phi(\mu_l)\right).$
    {Compute $\beta$ using line search as discussed in Subsection 2.3.1}
end for
```

- ***Testing Phase*** **(Algorithm 2).** In this phase, we use the μ^* learned from the *training* phase and testing data $\left\{f_{test}^{(i)}\right\}_{i=1}^{m_{test}}$ in Algorithm 2. The depth of the network at the *testing* phase is n_{test} layers. This phase can be carried out online, once the experimental data f_{test} becomes available.

Algorithm 2 Testing Phase of BONNet

```
Input: $\mu^*$, $\left\{f_{test}^{(i)}\right\}_{i=1}^{m_{test}}$, $m_{test}$ testing samples
Output: $u$
Initialize $u_0$
for for $j = 1$ to $n_{test}$ do
    Compute $u$ for all $i = 1, ..., m_{test}$:
        $$u_j^{(i)} = P_{X_{ad}}\Big(u_{j-1}^{(i)} - \frac{\alpha}{m}\big[K^*(Ku_{j-1}^{(i)} - f_{test}^{(i)}) + (\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma\big]\Big).$$
    {Compute $\alpha$ using line search as discussed in Subsection 2.3.1}
end for
```

Remark 2.3.2 (**Fixed vs. Variable Depth of BONNet**). We remark that instead of specifying the number of layers when solving (2.3.8) or (2.3.6), one could also, in addition, specify a stopping criterion appropriate for the solver being used, which is what we have done in our numerical examples. This is more in the spirit of solving an optimization problem which converges to a solution. The benefit of doing so is to prevent unnecessary computations, if the solver stopping criterion is reached earlier. This implies that the layers of the deep BONNet, in this case, will be variable. In our numerical experiments, we have used the stopping criterion for projected gradient descent method as mentioned in [93, pg. 91] for both μ and u. Also note that for (2.3.6), the number of layers in the testing phase (n_{test}) does not have to be equal to the number of layers in the training phase (n). In fact, $n << n_{test}$ prevents the network from *overfitting* of parameters to the training data, and helps the model generalize to unseen data [131]. Furthermore, reconstruction at the testing phase can be progressively improved for structural fidelity, if needed, by using a larger n_{test} (or a stricter stopping criterion). This allows for a trade-off between the quality of reconstruction and computational time.

2.3.1.2 BONNet Framework for Fractional Laplacian and Total Variation Regularization

In the general framework of our proposed deep BONNet, for any bounded linear operator K, any choice of regularizer can be incorporated, as long as it is cast into the generalized regularizer framework (2.3.2). In Section 2.2, we have proposed the use of fractional Laplacian as a regularizer, and have compared it with total variation regularization. We now show how to incorporate these regularizers into the deep BONNet, for a general K:

(a) **Fractional Laplacian Regularization.** Recall the fractional Laplacian regularization from (2.2.5),

$$\mathcal{R}(u,\mu) = \frac{1}{2}\|\sqrt{\lambda}(-\Delta)^{\frac{s}{2}}u\|^2_{L^2(\Omega)},$$

where $\mu = (\lambda, s)$ and $s \in (0,1)$. Then, to define the corresponding generalized regularizer (2.3.2), let $T(\mu, u(\mu)) := \sqrt{\lambda}(-\Delta)^{\frac{s}{2}}u$, and the activation function $\sigma(T) := T$. We omit the superscript 'i' to improve readability. Then, after some simplifications, (2.3.7), (2.3.11), and (2.3.12) become, for $j = 1, ..., n$,

$$u_j = P_{X_{ad}}\Big(u_{j-1} - \frac{\alpha}{m}\big[K^*(Ku_{j-1} - f) + \lambda(-\Delta)^s u_{j-1}\big]\Big),$$

$$\frac{\partial u_j}{\partial u_{j-1}} = I - \frac{\alpha}{m}K^*K - \frac{\alpha\lambda}{m}(-\Delta)^s,$$

and

$$\frac{\partial u_j}{\partial \lambda} = -\frac{\alpha}{m}(-\Delta)^s u_{j-1}, \quad \text{and} \quad \frac{\partial u_j}{\partial s} = -\frac{\alpha\lambda}{m}\frac{\partial}{\partial s}((-\Delta)^s u_{j-1}) \tag{2.3.13}$$

which together give us the sensitivity of u w.r.t. μ in (2.3.10). Notice that the second equation in (2.3.13) requires the sensitivity of fractional Laplacian $(-\Delta)^s$ with respect to 's'. This is a highly delicate object to handle. We shall reserve further details on this topic until the next section.

(b) **Total Variation Regularization.** Recall the total variation regularization

$$\mathcal{R}(u,\mu) = \lambda\,\mathrm{TV}_{\xi}(u),$$

where $\mu = \lambda$, and we are using the "regularized" total variation semi-norm,

$$\mathrm{TV}_{\xi}(u) = \int_{\Omega} \sqrt{|\nabla u|^2_{\ell^2(\Omega)} + \xi^2}\,\partial\Omega. \tag{2.3.14}$$

with $0 < \xi \ll 1$. We will omit the subscript ξ from TV_{ξ} for brevity. Then, to define the corresponding generalized regularizer (2.3.2), let $T(\mu, u(\mu)) := 2|\Omega|^{-1}\lambda\mathrm{TV}(u)$, and the activation function $\sigma(T) := \sqrt{T}$. Then, after some simplifications, (2.3.7), (2.3.11), and (2.3.12) become, for $j = 1, ..., n$,

$$\begin{aligned} u_j = P_{X_{ad}}\Big(u_{j-1} - \frac{\alpha}{m}\Big[K^*(Ku_{j-1} - f) + \\ \lambda\Big(-\operatorname{div}\big(\frac{\nabla u_{j-1}}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\big)\Big)\Big]\Big),\end{aligned}$$

$$\begin{aligned} \frac{\partial u_j}{\partial u_{j-1}} &= I - \frac{\alpha}{m}K^*K + \frac{\alpha\lambda}{2m}\operatorname{div}\Big(\frac{\partial}{\partial u_{j-1}}\big(\frac{\nabla u_{j-1}}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\big)\Big) \\ &= I - \frac{\alpha}{m}K^*K + \frac{\alpha\lambda}{2m}\operatorname{div}\Big(\frac{\nabla}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\Big) \\ &\quad + \frac{\alpha\lambda}{2m}\operatorname{div}\Big(\nabla u_{j-1}\frac{\partial}{\partial u_{j-1}}\big(\frac{1}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\big)\Big),\end{aligned} \tag{2.3.15}$$

and

$$\frac{\partial u_j}{\partial \lambda} = -\frac{\alpha}{2m}\Big(-\operatorname{div}\big(\frac{\nabla u_{j-1}}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\big)\Big)^*,$$

which together give us the sensitivity of u w.r.t. μ in (2.3.10). Again, we have omitted the superscript 'i' to improve readability.

2.4 Numerical Experiments of Tomographic Reconstruction

In this section, we present several numerical experiments where we apply our proposed BONNet to a tomographic reconstruction problem. We have introduced tomographic reconstruction in Subsection 2.2.3. We demonstrate the results of BONNet with two regularizers, namely, the total variation and the proposed fractional Laplacian.

All the computations are carried out using MATLAB R2015b on a Laptop with Intel Core i7-8550U Processor, with NVIDIA GeForce MX150 with 2 GB RAM. In view of Theorem 2.3.2, we run the proposed algorithm until a desired tolerance (tol) is met. At the testing phase we set $\text{tol} = 1 \times 10^{-5}$ and at the training phase we set $\text{tol} = 1 \times 10^{-3}$. Notice that the former is stricter than latter to avoid *overfitting*.

For all the total variation experiments we set the regularization parameter ξ in (2.3.14) as $\xi = 1 \times 10^{-5}$. In our numerical examples, we have noticed that the last term in (2.3.15) and the factor $\sqrt{(\cdot)}$ in the second last term does not play a significant role.

The remainder of the section is organized as follows. First in Subsection 2.4.1 we discuss the implementation details of fractional Laplacian and the admissible sets X_{ad} and $\mathcal{M}_{ad}$. This is followed by two experiments in Subsection 2.4.2.

2.4.1 Preliminaries

Before we discuss the actual results, we state some preliminary material. As mentioned in the paragraph following (2.2.7), we discretize Ω as $N \times N$ uniform pixels. Then given N_θ number of angles and N_τ number of discrete beamlets, our goal is to recover $u \in \mathbb{R}^{N^2}$. We also recall that the discrete form of operator K is the matrix $\mathbf{K} = (k_{i,j})_{i,j=1}^{N_\theta N_\tau, N^2}$. All the integrals are computed using uniform quadrature and the differential operators are discretized using finite differences. We shall discuss the approximation of fractional

Laplacian next.

2.4.1.1 Numerical Approximation of Fractional Laplacian

In order to approximate the fractional Laplacian, we first discretize the Laplacian $(-\Delta)$ on a uniform stencil. We denote the resulting discrete matrix by $\mathbf{A}$. If the eigen-decomposition of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1},$$

where $\mathbf{D} = (d_{i,j})_{i,j=1}^{N^2,N^2}$ with $d_{i,j} = 0$ if $i \neq j$, and $d_{i,i} = \zeta_i$ denotes the eigenvalues with columns of $\mathbf{V}$ containing the corresponding eigenvectors. Then the fractional power of $\mathbf{A}$ is given by,

$$\mathbf{A}^s = \mathbf{V}\mathbf{G}(s)\mathbf{V}^{-1},$$

where $\mathbf{G}(s) = (g_{i,j}(s))_{i,j=1}^{N^2,N^2}$ is the diagonal matrix with $g_{i,j}(s) = 0$ if $i \neq j$ and $g_{i,i}(s) = \zeta_i^s$. From (2.3.13) we also recall that we need to approximate the variation of $\mathbf{A}^s$ with respect to 's'. A straightforward calculation gives

$$\frac{d}{ds}\mathbf{A}^s = \mathbf{V}\mathbf{H}(s)\mathbf{V}^{-1}$$

where $\mathbf{H}(s) = (h_{i,j}(s))_{i,j=1}^{N^2,N^2}$ is the diagonal matrix with $h_{i,j}(s) = 0$ if $i \neq j$ and $h_{i,i}(s) = \zeta_i^s \ln(\zeta_i)$.

We remark that the scalability of numerical approximations of the fractional Laplacian can be handled using the approaches described in [16] and the references therein.

2.4.1.2 Admissible Sets and Projection

For tomographic reconstruction we let $X_{ad} := \{u \in X \mid u \geq 0\}$. Moreover, we set $\mathcal{M}_{ad} := \Lambda_{ad}$ for total variation and $\mathcal{M}_{ad} := \Lambda_{ad} \times S_{ad}$ where $\Lambda_{ad} := \{\lambda \in \mathbb{R} \mid \lambda \geq \epsilon_1 > 0\}$ and $S_{ad} := \{s \in \mathbb{R} \mid 0 < \epsilon_2 \leq s \leq 1 - \epsilon_2\}$. We let $\epsilon_1 = \epsilon_2 = 10^{-15}$.

Furthermore, the projection in (2.3.6) onto the admissible set X_{ad} is given by, for $z \in X$,

$$P_{X_{ad}}(z) := \max\{0, z\} = \begin{cases} z & \text{if} \quad z \geq 0, \\ \\ 0 & \text{if} \quad z < 0. \end{cases} \tag{2.4.1}$$

Formally, the "derivative" of this map is given by

$$\frac{d}{dt}\left(P_{X_{ad}}(z)\right) := \begin{cases} \dfrac{dz}{dt} & \text{if} \quad z \geq 0, \\ \\ 0 & \text{if} \quad z < 0. \end{cases}$$

For a rigorous definition of the generalized derivative of the *max* function, see [52]. Similar projection formulas are applicable for projection onto the set $\mathcal{M}_{ad}$.

2.4.1.3 Major Computational Costs

In Algorithm 1, two projected gradient descent schemes are being used to solve the outer and the inner level optimization problems. For each outer iteration, we solve the inner optimization problem, until convergence, using the projected gradient descent scheme. The convergence rate for the projected gradient descent method is well-known, see [93]. We elaborate on Step 4 of the algorithm. The two expensive components to compute $u_j^{(i)}$ are: (i) Evaluation of $K^*(Ku_{j-1}^{(i)} - f^{(i)})$, which at the discrete level requires 2 matrix vector multiplications and 1 subtraction; (ii) Evaluation of $(\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma$. Recall that for fractional Laplacian regularization, $T(\mu, u(\mu)) := \sqrt{\lambda}(-\Delta)^{\frac{s}{2}} u$ and $\sigma(T) = T$. Once $\mathbf{A}^{\frac{s}{2}}$ (similarly $\mathbf{A}^s$) has been pre-computed (see Subsection 2.4.1.1), the major computational cost associated with evaluation of $(\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma$ is one matrix vector multiplication $\mathbf{A}^s\mathbf{u}$.

The remainder of the cost in Step 4 is to evaluate the derivative of $u_j^{(i)}$ with respect to

μ. This can be done in an iterative fashion as described in the algorithm.

2.4.2 Experiments

We begin by generating the synthetic data. We create 30 distinct 64×64 samples (i.e. $N = 64$), which are variations of the Shepp-Logan Phantom (see Figure 2.3 for two representative samples). We use a convention of choosing $N_\tau \geq \sqrt{2}N$ beamlets. This choice ensures the maximum length of the 2D sample (i.e. its diagonal) is fully covered by the beamlets. Thus, for our experiments, we used $N_\tau = 93$. Then, for a given N_θ we simulate the corresponding sinogram f based on standard discrete Radon transform [26]. Next we add 0.1% Gaussian noise to each sinogram, respectively. This gives us our synthetic data, which we divide into $m = 20$ training samples and $m_{test} = 10$ testing samples.

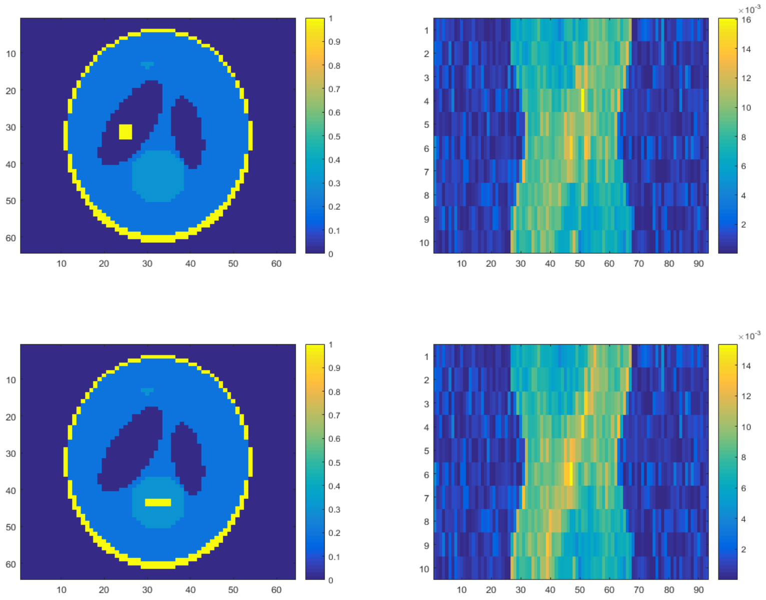

Figure 2.3: Representative samples of Phantom (u_{true}) used (*left*) to generate the synthetic data (noisy sinogram f) (*right*) for training (*Row 1*) and testing (*Row 2*).

We remark that in tomography, the *number of projection angles*, N_θ, has a significance, since it determines the amount of X-ray the sample is exposed to. We emphasize that the most challenging, yet common, cases in tomographic reconstruction are the ones with smaller N_θ, due to the limits on X-ray exposure. We conduct numerical experiments for tomographic scans obtained for various N_θ. For each choice, the selected number of angles are uniformly distributed in the range $[0, 180]$. Note that, for each choice of N_θ, a separate set of projection data is generated (for a batch of 30 samples), on which the learning and reconstructions are performed using our deep BONNet as discussed in Algorithm 1 and Algorithm 2.

We have undertaken two sets of experiments. In the first experiment, we fix $s = 0.4$ and learn λ. In the second experiment, we learn $\mu = (\lambda, s)$.

2.4.2.1 Results of Experiment I: Learning λ, fixed $s = 0.4$

We now discuss the results of our experiments. In Figure 2.4, we compare the reconstructions obtained from BONNet with the true solution shown in Figure 2.3. The reconstructions are based on 'no regularization', total variation regularization, and the fractional Laplacian regularization for data with 0.1% noise. The columns correspond to the number of projections angles used. We remark again that each choice of N_θ for a batch of training and testing data, corresponds to a distinct separate problem that we solve, as the dimensionality of $\mathbf{K}$ depends on N_θ. The *left panel* corresponds to the reconstruction of the *training* data at the nth iterate. Recall that at the training phase, $\{(u_{true}^{(i)}, f_{train}^{(i)})\}_{i=1}^{m=20}$ are passed to the deep BONNet Algorithm 1. The λ values mentioned under each reconstruction are the corresponding optimal λ^*_{none}, λ^*_{TV}, and $\lambda^*_{fracLap}$ that we learn during the training stage. Notice that $\lambda^*_{none} = 0$ corresponds to 'no regularization'. The *right panel* corresponds to the reconstructions at the $n_{test}th$ layer of the testing phase. Recall that $\{(\lambda^*, f_{test}^{(i)})\}_{i=1}^{m_{test}=10}$ are passed to the deep BONNet at this stage in Algorithm 2.

From the reconstructions in Figure 2.4, we observe that for the tomographic reconstruction problem, first of all, regularization is improving the quality of reconstructions. In the absence of regularization, the high intensity regions are preserved, but we lose information

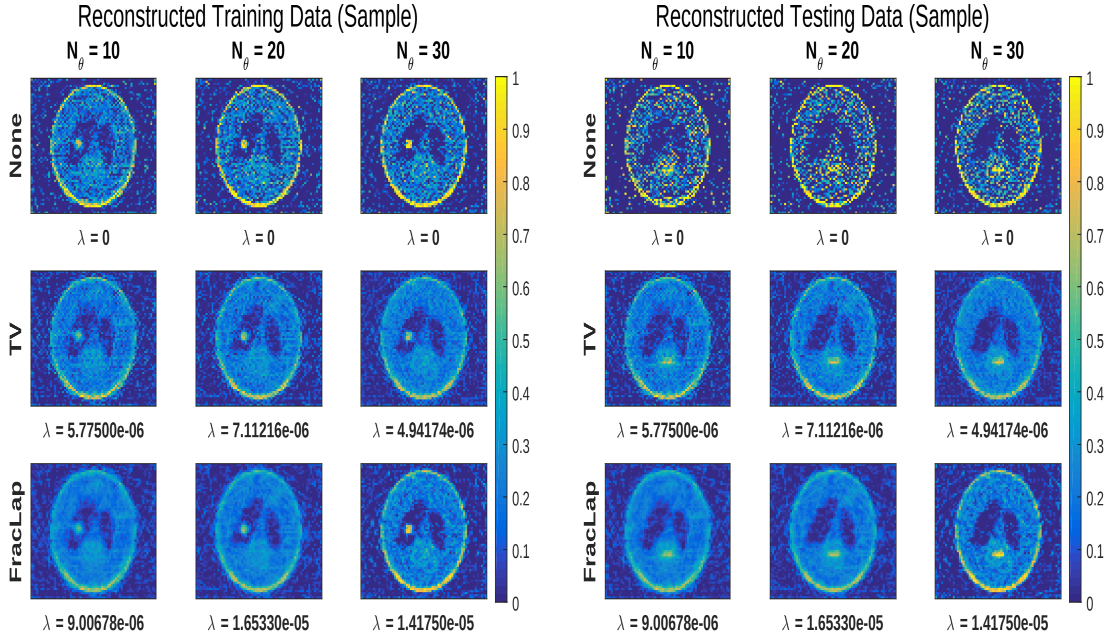

Figure 2.4: Comparison of reconstructions based on various regularizers (*rows*) and various number of tomographic projection angles (*columns*) for data with 0.1% Gaussian noise. The *left* and *right* panels correspond to the solution at the last layer for two of the many distinct samples used during training and testing phases, respectively. The λ values mentioned are the optimal values obtained from the deep BONNet training, which are then used for the reconstructions during the corresponding testing phase.

from regions of low intensity. On the other hand, TV and fractional Laplacian regularizations preserve the sample characteristics in the lower intensity regions of the sample. Fractional Laplacian gives reconstructions which are either better, or comparable to TV regularization. In addition, it does better at smoothing out the noise, and also in regaining comparatively more information in regions of low intensity, such as the dim circle on the lower side of the Phantom, e.g. for $N_\theta = 10$. This is especially important when we have limited data to reconstruct from. We also recall that the Euler-Lagrange equation corresponding to the fractional Laplacian regularization is linear, and that of TV is non-linear.

We also observe that for any given regularizer choice, the optimal λ^* obtained for

$N_\theta = 10$ is similar to the one obtained for a larger N_θ. Thus, to learn the regularization strength, even limited tomographic scan data suffices, and the same λ^* could be used for reconstruction at the testing phase for any amount of available data, which can significantly save the offline training time.

For the experimental cases mentioned above, we measure the quality of reconstructions using metrics such as the *mean-squared error* (MSE) Figure 2.5, *Peak signal-to-noise ratio* (PSNR) Figure 2.6, and *structural similarity index* (SSIM) Figure 2.7, averaged over all the samples. For MSE, smaller values correspond to better results, and for PSNR and SSIM, larger values are better. Notice that for each metric, fractional Laplacian regularization outperforms the total variation regularization.

We remark that the λ values that we learn via deep BONNet are similar to those obtained by using a combination of the lowest error norm and L-curve; however, the parameter search via BONNet is automated. The reconstructions obtained via Projected Gradient Descent are also similar to the ones obtained earlier Figure 2.2 using the inexact truncated-Newton method for bound-constrained problem [111]. We emphasize that one may use a different solver during the testing stage once λ^* is obtained via BONNet training.

2.4.2.2 Results of Experiment II: Learning λ and Fractional Exponent 's'

We now train BONNet to learn both the fractional exponent 's' of the fractional Laplacian and the strength λ. We use the BONNet architecture using fractional Laplacian discussed in Subsection 2.3.1.2 and use the same training and testing data as described in the previous example. In Table 2.1 we show comparisons of MSE, SSIM and PSNR for $N_\theta = \{10, 20\}$ projection angles, respectively, for the reconstructions of the testing data. We compare the results with the fractional Laplacian case discussed in Subsection 2.4.2.1. In the case of $N_\theta = 10$, we obtain $(\lambda^*_{fracLap}, s^*) = (5.04417\text{e-}6,\ \ 0.5413)$ and in the case of $N_\theta = 20$, we obtain $(\lambda^*_{fracLap}, s^*) = (8.53717\text{e-}6,\ \ 0.3799)$. The reconstructions of u with $(\lambda^*_{fracLap}, s^*)$ are visually comparable to the case of fractional Lapalcian in Figure 2.4 and therefore they have been omitted. We observe that all the error metrics returned by BONNet are either

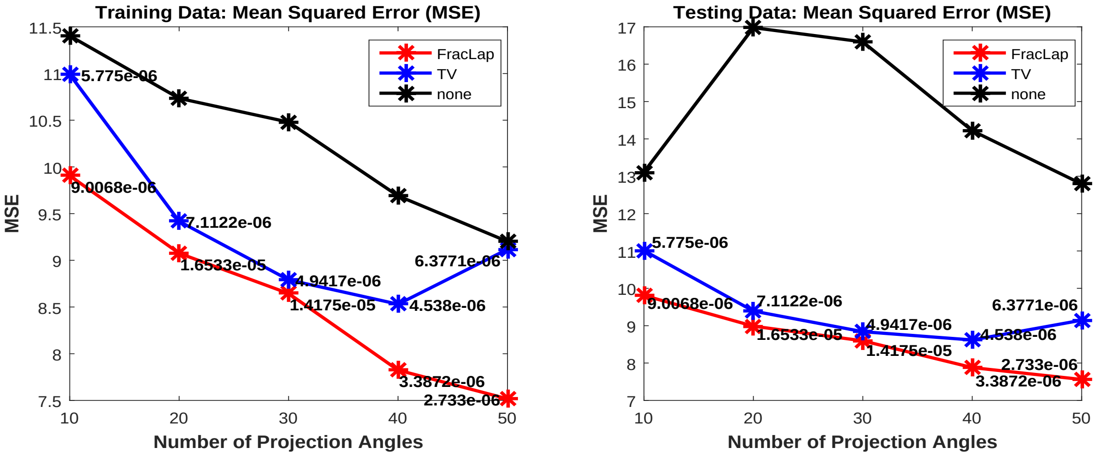

Figure 2.5: We compare the mean-squared errors (MSE) for the solution, averaged over 20 training (respectively, 10 testing) samples (*left*(respectively, right)), against various number of projection angles for the tomographic reconstruction problem. The solid *black, blue* and *red* lines corresponds to 'no regularization', total variation regularization, and fractional Laplacian regularization, respectively. For each experiment, the λ^* learned from BONNet at the training phase is mentioned, which is in turn used for the reconstruction during training (*left*) and testing (*right*) phases. Smaller values of MSE correspond to better results, and fractional Laplacian outperforms the others. Note that 0.1% Gaussian noise was added to the data 'f', and $s = 0.4$ for fractional Laplacian.

comparable, or slightly better, than the ones obtained by BONNet for a fixed 's', discussed in Subsection 2.4.2.1. The advantage now is that we no longer need to tune the parameters manually.

2.5 Discussion

In this work, we consider a general regularized regression model for inverse problems. This model can incorporate the underlying physics (defined by the operator K), in addition to the prior knowledge of the solution in the regularization term. However, to fully explore the potential of this generalized model, an optimal choice of the type of regularizer, as well as the regularization strength, is inevitable.

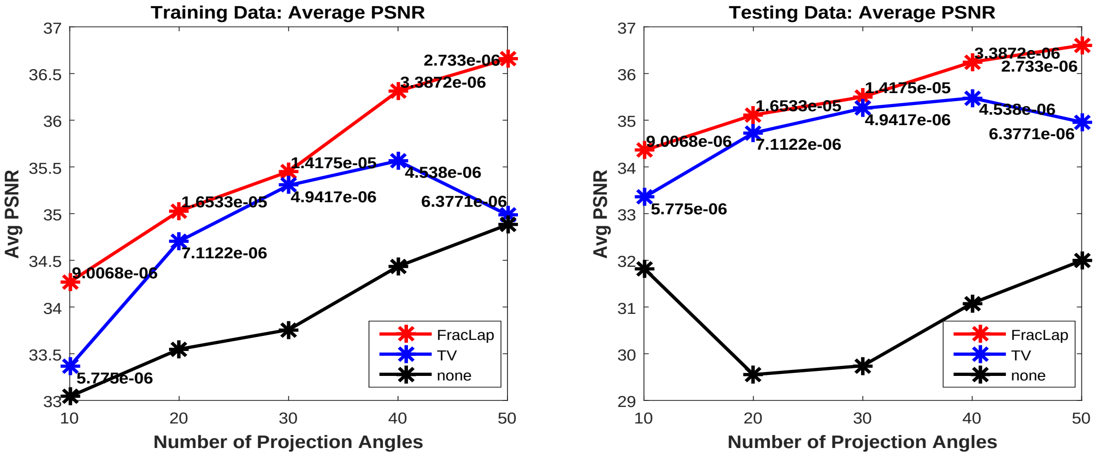

Figure 2.6: We compare the peak signal-to-noise ratio (PSNR) for the solution, averaged over 20 training (respectively, 10 testing) samples (*left*(respectively, right)), against various number of projection angles for the tomographic reconstruction problem. The solid *black, blue* and *red* lines corresponds to 'no regularization', total variation regularization, and fractional Laplacian regularization, respectively. For each experiment, the λ^* learned from BONNet at the training phase is mentioned, which is in turn used for the reconstruction during training (*left*) and testing (*right*) phases. Larger values of PSNR correspond to better results, and fractional Laplacian outperforms the others. Note that 0.1% Gaussian noise was added to the data 'f', and $s = 0.4$ for fractional Laplacian.

Table 2.1: Comparison of average MSE, SSIM and PSNR for tomographic reconstructions obtained via BONNet using the fractional Laplacian regularization for two distinct number of projection angles. In Experiment I, we fix $s = 0.4$ and learn λ^* via BONNet, and in Experiment II we learn the (λ^*, s^*) pair. The results shown are for the testing dataset. Notice that the search for $\mu^* = (\lambda^*, s^*)$ in Experiment II is now fully automated and the results are better or comparable to Experiment I.

Data	**Testing**			
N_θ	10		20	
Type	Experiment I	Experiment II	Experiment I	Experiment II
(λ, s)	$(9.00678e-6, 0.4)$	$(5.04417e-6, 0.5413)$	$(1.65330e-5, 0.4)$	$(8.53717e-6, 0.3799)$
MSE	9.8099	9.7743	8.9872	8.6961
SSIM	0.7675	0.7738	0.7888	0.7950
PSNR	34.3513	34.3831	35.1123	35.3973

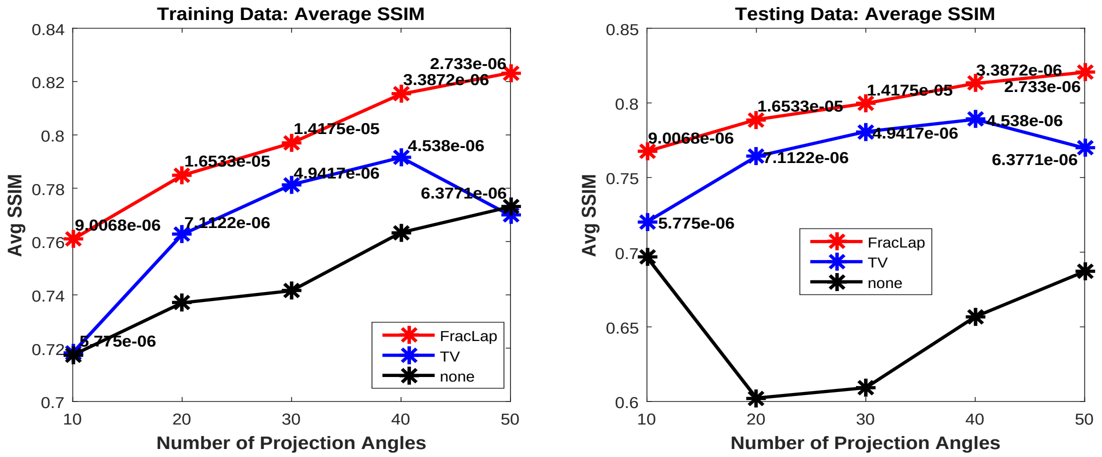

Figure 2.7: We compare the peak structural similarity (SSIM) for the solution, averaged over 20 training (respectively, 10 testing) samples (*left*(respectively, right)), against various number of projection angles for the tomographic reconstruction problem. The solid *black, blue* and *red* lines corresponds to 'no regularization', total variation regularization, and fractional Laplacian regularization, respectively. For each experiment, the λ^* learned from BONNet at the training phase is mentioned, which is in turn used for the reconstruction during training (*left*) and testing (*right*) phases. Larger values of SSIM correspond to better results, and fractional Laplacian outperforms the others. Note that 0.1% Gaussian noise was added to the data 'f', and $s = 0.4$ for fractional Laplacian.

We have used fractional Laplacian as a regularizer on tomographic reconstruction problems. Previously, this has been used in image denoising. The key benefit of using this regularization is that the corresponding Euler-Lagrange equation is *linear*, as opposed to the *nonlinear* and possibly *degenerate* Euler-Lagrange equation for the popular total variation regularization.

To address the challenge of finding the optimal regularization strength, we introduce a dedicated deep BONNet architecture to learn the regularization parameters for any choice of regularizer. We show an analogy of the regularization function to the activation function in a standard neural network, which provides a theoretical guidance in terms of choosing an optimal activation function. In addition to the regularization strength λ, BONNet can also learn the exponent 's' for the fractional Laplacian regularization.

Next, we demonstrate the benefit of our proposed deep BONNet on the tomographic reconstruction problem. We first conduct experiments to learn only λ with a fixed 's'. We have observed that fractional Laplacian regularization gives comparable or better reconstructions compared to the total variation regularization. Especially for the noisy and limited data ($N_\theta = 10$), fractional Laplacian regularization outperforms the total variation regularization. In contrast to the standard machine learning architectures with fixed number of layers, our network favors a variable number of layers (depth) which is dictated by the convergence to the solution of the optimization problem. Thus, the number of layers in the network can be different for different samples and different regularizers. We also demonstrate the capability of our proposed BONNet in terms of learning the optimum $(\lambda^*_{fracLap}, s^*)$ pair for the fractional Laplacian regularizer, and this indicates the flexibility of our proposed network to learn non-standard parameters.

Chapter 3: Fractional Deep Neural Network via Constrained Optimization [1]

3.1 Introduction

Deep learning has emerged as a potent area of research and has enabled a remarkable progress in recent years spanning domains like imaging science [8, 83, 91, 152], biomedical applications [48, 78, 100], satellite imagery, remote sensing [35, 137, 155], etc. However, the mathematical foundations of many machine learning architectures are largely lacking [61, 67, 117, 124, 150]. The current trend of success is largely due to the empirical evidence. Due to the lack of mathematical foundation, it becomes challenging to understand the detailed workings of networks [70, 108].

The overarching goal of machine learning algorithms is to learn a function using some known data. Deep Neural Networks, like Residual Neural Networks, are a popular family of deep learning architectures which have turned out to be groundbreaking in imaging science. An introductory example of RNN is the ResNet [83] which has been successful for classification problems in imaging science. Compared to the classical DNNs, the innovation of the RNN architecture comes from a simple addition of an identity map between each layer of the network. This ensures a continued flow of information from one layer to another. Despite their success, DNNs are prone to various challenges such as vanishing gradients [30, 67, 142], difficulty in approximating non-smooth functions, long training time [47], etc.

We remark that recently in [88] the authors have introduced a DenseNet, which is a new approach to prevent the gradient “wash out” by considering dense blocks, in which each layer takes into account all the previous layers (or the memory). They proceed by concatenating

[1] This work is under review [9].

the outputs of each dense block which is then fed as an input to the next dense block. Clearly as the number of layers grow, it can become prohibitively expensive for information to propagate through the network. DenseNet can potentially overcome the vanishing gradient issue, but it is only an adhoc method [88, 156]. Some other networks that have attempted to induce multilayer connections are Highway Net [133], AdaNet [54], ResNetPlus [49], etc. All these models, however, largely lack rigorous mathematical frameworks. Furthermore, rigorous approaches to learn nonsmooth functions such as the absolute value function $|x|$ are scarce [89].

There has been a recent push in the scientific community to develop rigorous mathematical models and understanding of the DNNs [61]. One way of doing so is to look at their architecture as dynamical systems. The articles [31, 76, 104, 124, 127] have established that a DNN can be regarded as an optimization problem subject to a discrete ordinary differential equation as constraints. The limiting problem in the continuous setting is an ODE constrained optimization problem [124, 127]. Notice that designing the solution algorithms at the continuous level can lead to architecture independence, i.e., the number of iterations remains the same even if the number of layers is increased.

The purpose of this work is to present a novel fractional deep neural network which allows the network to access historic information of input and gradients across all subsequent layers. This is facilitated via our proposed use of fractional derivative based ODE as constraints. We derive the optimality conditions for this network using the Lagrangian approach. Next, we consider a discretization for this fractional ODE and the resulting DNN is called *Fractional-DNN*. We provide the algorithm and show numerical examples on some standard datasets.

For completeness, we also mention the Fractional Physics Informed Neural Networks (fPINNs) [115] where the authors aim to solve Partial Differential Equations (PDEs), in particular fractional PDEs. This is an extension of authors' previous works in [75, 119]. The idea is to minimize the PDE residual in a least-squares formulation and learn the unknown parameters using a standard feedforward NN. This is completely different than what we are proposing in this paper. Our goal is to introduce a new NN with memory using the fractional derivatives. We formulate the NN problem as an ODE constrained optimal control problem and use Lagrangian formulation to derive the optimality conditions. We apply our NN to classification problems but notice that it can also be applied to the problems discussed in [115, 119].

Owing to the fact that fractional time derivatives allow memory effects, in the Fractional-DNN all the layers are connected to one another, with an appropriate scaling. In addition, fractional time derivatives can be applied to nonsmooth functions [12]. Thus, we aim to keep the benefits of standard DNN and the ideology of DenseNet, but remove the bottlenecks.

The learning rate in a neural network is an important hyper-parameter which influences training [29]. In our numerical experiments, we have observed an improvement in the learning rate via Fractional-DNN, which enhances the training capability of the network. Our numerical examples illustrate that, Fractional-DNN can potentially solve the vanishing gradient issue (due to memory), and handle nonsmooth data.

This Chapter is organized as follows. In Section 3.2 we introduce notations and definitions. We introduce our proposed Fractional-DNN in Section 3.3. This is followed by Section 3.4 where we discuss its numerical approximation. In Section 3.5, we state our algorithm. The numerical examples given in Section 3.6 show the working and improvements due to the proposed ideas on three different datasets.

3.2 Preliminaries

The purpose of this section is to introduce some notations and definitions that we will use throughout this Chapter. We begin with Table 3.1 where we state the standard notations. In Subsection 3.2.1 we describe the well-known softmax loss function. Subsection 3.2.2 is dedicated to the Caputo fractional time derivative.

Table 3.1: Table of Notations.

Symbol	Description
$n \in \mathbb{N}$	Number of distinct samples
$n_f \in \mathbb{N}$	Number of sample features
$n_c \in \mathbb{N}$	Number of classes
$N \in \mathbb{N}$	Number of network layers (i.e. network depth)
$Y \in \mathbb{R}^{n_f \times n}$	$Y = \{y^{(i)}\}_{i=1}^{n}$ is the collective feature set of n samples.
$C_{obs} \in \mathbb{R}^{n_c \times n}$	$C_{obs} = \{c^{(i)}\}_{i=1}^{n}$ are the true class labels of the input data
$W \in \mathbb{R}^{n_c \times n_f}$	Weights
$K \in \mathbb{R}^{n_f \times n_f}$	Linear operator (distinct for each layer)
$b \in \mathbb{R}$	Bias (distinct for each layer)
$P \in \mathbb{R}^{n_f \times n}$	Lagrange multiplier
$e_{n_c} \in \mathbb{R}^{n_c}$	A vector of ones
$\tau \in \mathbb{R}$	Time step-length
$\sigma(\cdot)$	Activation function, acting pointwise
γ	Order of fractional time derivative
$(\cdot)'$	Derivative w.r.t. the argument
$tr(\cdot)$	Trace operator
$(\cdot)^\intercal$	Matrix transpose
$\odot$	Point-wise multiplication
m_1	Max count for randomly selecting a mini-batch in training
m_2	Max iteration count for gradient-based optimization solver
$\alpha_{train},\ \alpha_{test}$	Percentage of training and testing data correctly identified

3.2.1 Cross Entropy with Softmax Function

Given collective feature matrix Y with true labels C_{obs} and the unknown weights W, the cross entropy loss function given by

$$E(W, Y, C_{obs}) = -\frac{1}{n}\,\mathrm{tr}(C_{obs}^\intercal\ \log(S(W, Y))) \tag{3.2.1}$$

measures the discrepancy between the true labels C_{obs} and the predicted labels $\log(S(W, Y))$. Here,

$$S(W, Y) := \exp(WY)\,\mathrm{diag}\left(\frac{1}{e_{n_c}^\intercal \exp(WY)}\right) \tag{3.2.2}$$

is the softmax classifier function, which gives normalized probabilities of samples belonging to the classes.

3.2.2 Caputo Fractional Derivative

In this section, we define the notion of Caputo fractional derivative and refer [12] and references therein for the following definitions.

Definition 3.2.1 (Left Caputo Fractional Derivative)**.** For a fixed real number $0 < \gamma < 1$, and an absolutely continuous function $u\colon [0,T] \to \mathbb{R}$, the left Caputo fractional derivative is defined by:

$$d_t^\gamma u(t) = \frac{1}{\Gamma(1-\gamma)} \frac{d}{dt} \int_0^t \frac{u(r) - u(0)}{(t-r)^\gamma} \, dr, \tag{3.2.3}$$

where $\Gamma(\cdot)$ is the Euler-Gamma function.

Definition 3.2.2 (Right Caputo Fractional Derivative)**.** For a fixed real number $0 < \gamma < 1$, and an absolutely continuous function $u\colon [0,T] \to \mathbb{R}$, the right Caputo fractional derivative is defined by:

$$d_{T-t}^\gamma u(t) = \frac{-1}{\Gamma(1-\gamma)} \frac{d}{dt} \int_t^T \frac{u(r) - u(T)}{(r-t)^\gamma} \, dr. \tag{3.2.4}$$

Notice that, $d_t^\gamma u(t)$ and $d_{T-t}^\gamma u(t)$ in definitions (3.2.3) and (3.2.4) exist almost everywhere on $[0,T]$, [94, Theorem 2.1], and are represented, respectively, by

$$d_t^\gamma u(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{u'(r)}{(t-r)^\gamma} \, dr, \qquad \text{and} \qquad d_{T-t}^\gamma u(t) = \frac{-1}{\Gamma(1-\gamma)} \int_t^T \frac{u'(r)}{(r-t)^\gamma} \, dr.$$

Moreover, if $\gamma = 1$ and $u \in C^1([0,T])$, then one can show that $d_t^\gamma u(t) = u'(t) = d_{T-t}^\gamma u(t)$. We note that the fractional derivatives in (3.2.3) and (3.2.4) are nonlocal operators. Indeed, the derivative of u at a point t depends on all the past and future events, respectively. This behavior is different than the classical case of $\gamma = 1$.

The left and right Caputo fractional derivatives are linked by the fractional integration by parts formula, [15, Lemma 3], which will be stated next. For $\gamma \in (0,1)$, let

$$\mathbb{L}_\gamma := \left\{ f \in C([0,T]) : d_t^\gamma f \in L^2(0,T) \right\}, \qquad \mathbb{R}_\gamma := \left\{ f \in C([0,T]) : d_{T-t}^\gamma f \in L^2(0,T) \right\}.$$

Lemma 3.2.3 (Fractional Integration-by-Parts)**.** For $f \in \mathbb{L}_\gamma$ and $g \in \mathbb{R}_\gamma$, the following integration-by-parts formula holds:

$$\int_0^T d_t^\gamma f(t) g(t) \, dt = \int_0^T f(t) d_{T-t}^\gamma g(t) \, dt + g(T)(I_t^{1-\gamma} f)(T) - f(0)(I_{T-t}^{1-\gamma} g)(0), \qquad (3.2.5)$$

where $I_t^{1-\gamma} w(t)$ and $I_{T-t}^{1-\gamma} w(t)$ are the left and right Riemann-Liouville fractional integrals of order γ and are given by

$$I_t^{1-\gamma} w(t) := \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{w(r)}{(t-r)^\gamma} \, dr \;\text{ and }\; I_{T-t}^{1-\gamma} w(t) := \frac{1}{\Gamma(1-\gamma)} \int_t^T \frac{w(r)}{(r-t)^\gamma} \, dr.$$

3.3 Continuous Fractional Deep Neural Network

After the above preparations, in this section, we shall introduce the Fractional-DNN. First we briefly describe the classical RNN, and then extend it to develop the Fractional-DNN. We formulate our problem as a constrained optimization problem. Subsequently, we shall use the Lagrangian approach to derive the optimality conditions.

3.3.1 Classical RNN

Our goal is to approximate a map $\mathcal{F}$. A classical RNN helps approximate $\mathcal{F}$, for a known set of inputs and outputs. To construct an RNN, for each layer j, we first consider a linear-transformation of Y_{j-1} as,

$$\mathcal{G}_{j-1}(Y_{j-1}) = K_{j-1} Y_{j-1} + b_{j-1},$$

where the pair (K_j, b_j) denotes an unknown linear operator and bias at the j^{th} layer. When $N > 1$ then the network is considered “deep”. Next we introduce non linearity using a nonlinear activation function σ (e.g. ReLU or tanh). The resulting RNN is,

$$Y_j = Y_{j-1} + \tau(\sigma \circ \mathcal{G}_{j-1})(Y_{j-1}), \quad j = 1, \cdots, N; \quad N > 1, \tag{3.3.1}$$

where $\tau > 0$ is the time-step. Finally, the RNN approximation of $\mathcal{F}$ is given by,

$$\mathcal{F}_\theta(\cdot) = \Big(\big(I + \tau(\sigma \circ \mathcal{G}_{N-1})\big) \circ \big(I + \tau(\sigma \circ \mathcal{G}_{N-2})\big) \circ \cdots \circ \big(I + \tau(\sigma \circ \mathcal{G}_0)\big)\Big)(\cdot),$$

with $\theta = (K_j, b_j)$ as the unknown parameters. In other words, the problem of approximating $\mathcal{F}$ using classical RNN, intrinsically, is a problem of learning (K_j, b_j).

Hence, for given datum (Y_0, C), the learning problem then reduces to minimizing a loss function $\mathcal{J}(\theta, (Y_N, C))$, subject to constraint (3.3.1), i.e.,

$$\begin{aligned} &\min_\theta \quad \mathcal{J}(\theta, (Y_N, C)) \\ &\text{s.t.} \quad Y_j = Y_{j-1} + \tau(\sigma \circ \mathcal{G}_{j-1})(Y_{j-1}), \quad j = 1, \ldots, N. \end{aligned} \tag{3.3.2}$$

Notice that the system (3.3.1) is the forward-Euler discretization of the following continuous in time ODE, see [68, 83, 124],

$$\begin{aligned} d_t Y(t) &= \sigma(K(t)Y(t) + b(t)), \quad t \in (0, T), \\ Y(0) &= Y_0. \end{aligned} \tag{3.3.3}$$

The continuous learning problem then requires minimizing the loss function $\mathcal{J}$ at the final time T subject to the ODE constraints (3.3.3):

$$\begin{aligned} \min_{\theta=(K,b)} \quad & \mathcal{J}(\theta,(Y(T),C)) \\ \text{s.t.} \quad & (3.3.3) \end{aligned} \tag{3.3.4}$$

Notice that designing algorithms for the continuous in time problem (3.3.4) instead of the discrete in time problem (3.3.2) has several key advantages. In particular, it will lead to algorithms which are independent of the neural network architecture, i.e., independent of the number of layers. In addition, the approach of (3.3.4) can help us determine the stability of the neural network (3.3.2), see [31,76]. Moreover, for the neural network (3.3.2), it has been noted that as the information about the input or gradient passes through many layers, it can vanish and "wash out", or grow and "explode" exponentially [30]. There have been adhoc attempts to address these concerns, see for instance [54, 88, 133], but a satisfactory mathematical explanation and model does not currently exist. One of the main goals of this work is to introduce such a model.

Notice that (3.3.3), and its discrete version (3.3.1), incorporates many algorithmic processes such as linear solvers, preconditioners, nonlinear solvers, optimization solvers, etc. Furthermore, there are well-established numerical algorithms that re-use information from previous iterations to accelerate convergence, e.g. the BFGS method [114], Anderson acceleration [5], and variance reduction methods [121]. These methods account for the history $Y_j, Y_{j-1}, Y_{j-2}, \ldots, Y_0$, while choosing Y_{j+1}. Motivated by these observations we introduce versions of (3.3.1) and (3.3.3) that can account for history (or memory) effects in a rigorous mathematical fashion.

3.3.2 Continuous Fractional-DNN

The fractional time derivative in (3.2.3) has a distinct ability to allow a memory effect, for instance in materials with hereditary properties [40]. Fractional time derivative can be derived by using the anomalous random walks where the walker experiences delays between jumps [110]. In contrast, the standard time derivative naturally arises in the case of classical random walks. We use the idea of fractional time derivative to enrich the constraint optimization problem (3.3.4), and subsequently (3.3.2), by replacing the standard time derivative d_t by the fractional time derivative d_t^γ of order $\gamma \in (0,1)$. Recall that for $\gamma = 1$, we obtain the classical derivative d_t. Our new continuous in time model, the Fractional-DNN, is then given by (cf. (3.3.3)),

$$\begin{aligned} d_t^\gamma Y(t) &= \mathcal{F}_\theta(Y(t), t, \theta(t)), \quad t \in (0,T), \\ Y(0) &= Y_0 \end{aligned} \tag{3.3.5}$$

where d_t^γ is the Caputo fractional derivative as defined in (3.2.3). The discrete formulation of Fractional-DNN will be discussed in the subsequent section.

The main reason for using the Caputo fractional time derivative over its other counterparts such as the Riemann Liouville fractional derivative is the fact that the Caputo derivative of a constant function is zero and one can impose the initial conditions $Y(0) = Y_0$ in a classical manner [125]. Note that d_t^γ is a nonlocal operator in a sense that in order to evaluate the fractional derivative of Y at a point t, we need the cumulative information of Y over the entire sub-interval $[0, t)$. This is how the Fractional-DNN enables connectivity across all antecedent layers (hence the memory effect). As we shall illustrate with the help of a numerical example in Section 3.6, this feature can help overcome the vanishing gradient issue, as the cumulative effect of the gradient of the precedent layers is less likely to be zero.

Remark 3.3.1 (Caputo Derivative of Nonsmooth Functions)**.** The Caputo fractional derivative (3.2.3) can be applied to non-smooth functions. Consider, e.g. $Y(t) := |t|$. Notice that $Y(t)$ is not differentiable at $t = 0$. However, (3.2.3) yields, $d_t^\gamma Y(t) = \frac{1}{\Gamma(2-\gamma)} t^{1-\gamma}$. Since $\gamma \in (0,1)$, therefore $d_t^\gamma Y(t)$ at $t = 0$ is zero. □

Owing to Theorem 3.3.1 we can better account for features, Y, which are non-smooth, as a result of which the smoothness requirement on the unknown parameters θ can be weakened. This, in essence, can help with the exploding gradient issue in DNNs.

The generic learning problem with Fractional-DNN as constraints can be expressed as,

$$\begin{aligned} \min_{\theta=(K,b)} \quad & \mathcal{J}(\theta,(Y(T),C)) \\ \text{s.t.} \quad & (3.3.5) \end{aligned} \tag{3.3.6}$$

Note that the choice of $\mathcal{J}$ depends on the type of learning problem. We will next consider a specific structure of $\mathcal{J}$ given by the cross entropy loss functional, defined in (3.2.1).

3.3.3 Continuous Fractional-DNN and Cross Entropy Loss Functional

Supervised learning problems are a broad class of machine learning problems which use labeled data. These problems are further divided into two types, namely regression problems and classification problems. The specific type of the problem dictates the choice of $\mathcal{J}$ in (3.3.6). Regression problems often occur in physics informed models, e.g. sample reconstruction inverse problems [8, 78]. On the other hand, classification problems occur, for instance, in computer vision [51, 126]. In both the cases, a neural network is used to learn the unknown parameters. In the discussion below we shall focus on classification problems, however, the entire discussion directly applies to regression type problems.

Recall that the cross entropy loss functional E, defined in (3.2.1), measures the discrepancy between the actual and the predicated classes. Replacing, $\mathcal{J}$ in (3.3.6) by E together with a regularization term $\mathcal{R}(W, K(t), b(t))$, we arrive at

$$
\begin{aligned}
&\min_{W,K,b} \quad E(W, Y(T), C_{obs}) + \mathcal{R}(W, K(t), b(t)) \\
&\text{s.t.} \quad \begin{cases} d_t^\gamma Y(t) = \sigma(K(t)Y(t) + b(t)), & t \in (0, T), \\ Y(0) = Y_0 \,. \end{cases}
\end{aligned}
\tag{3.3.7}
$$

Note that, in this case, the unknown parameter $\theta := (W, K, b)$, where K and b are, respectively, the linear operator and bias for each layer, and the weights W are a feature-to-class map. Furthermore, σ is a nonlinear activation function and (Y_0, C_{obs}) is the given data, with C_{obs} as the true labels of Y_0.

To solve (3.3.7), we rewrite this problem as an unconstrained optimization problem via the Lagrangian functional and derive the optimality conditions. Let P denote the Lagrange multiplier, then the Lagrangian functional is given by,

$$
\mathcal{L}(Y, W, K, b; P) := E(W, Y(T), C_{obs}) + \mathcal{R}(W, K(t), b(t)) + \langle d_t^\gamma Y(t) - \sigma(K(t)Y(t) + b(t)), P(t)\rangle,
$$

where, $\langle \cdot, \cdot \rangle := \int_0^T \langle \cdot, \cdot \rangle_F \, dt$ is the L^2-inner product, and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. Using the fractional integration-by-parts from (3.2.5), we obtain

$$
\begin{aligned}
\mathcal{L}(Y, W, K, b; P) = E(W, Y(T), C_{obs}) + \mathcal{R}(W, K(t), b(t)) - \langle \sigma(K(t)Y(t) + b(t)), P(t)\rangle \\
+ \langle Y(t), d_{T-t}^\gamma P(t)\rangle + \langle (I_t^{1-\gamma} Y)(T), P(T)\rangle_F - \langle Y_0, (I_{T-t}^{1-\gamma} P)(0)\rangle_F.
\end{aligned}
\tag{3.3.8}
$$

Let $(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P})$ denote a stationary point, then the first order necessary optimality conditions are given by the following set of state, adjoint and design equations:

(A) **State Equation.** The gradient of $\mathcal{L}$ with respect to P at $(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P})$ yields the state equation $\nabla_P \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P}) = 0$, equivalently,

$$\begin{aligned} d_t^\gamma \overline{Y}(t) =&\ \sigma(\overline{K}(t)\overline{Y}(t) + \overline{b}(t)), \qquad t \in (0,T), \\ \overline{Y}(0) =&\ Y_0 \end{aligned} \tag{3.3.9}$$

where d_t^γ denotes the left Caputo fractional derivative (3.2.3). In (3.3.9), for the state variable $\overline{Y}$, we solve forward in time, therefore we call (3.3.9) as the *forward propagation.*

(B) **Adjoint Equation.** Next, the gradient of $\mathcal{L}$ with respect to Y at $(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P})$ yields the adjoint equation $\nabla_Y \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P}) = 0$, equivalently,

$$\begin{aligned} d_{T-t}^\gamma \overline{P}(t) &= (\sigma'(\overline{K}(t)\overline{Y}(t) + \overline{b}(t))\ \overline{K}(t))^\intercal\ \overline{P}(t) \\ &= \overline{K}(t)^\intercal \left(\overline{P}(t) \odot \sigma'\left(\overline{K}(t)\overline{Y}(t) + \overline{b}(t)\right)\right), \qquad t \in (0,T), \\ \overline{P}(T) &= -\frac{1}{n}\overline{W}^\intercal(-C_{obs} + S(\overline{W}, \overline{Y}(T))) \end{aligned} \tag{3.3.10}$$

where d_{T-t}^γ denotes the right Caputo fractional derivative (3.2.4) and S is the softmax function defined in (3.2.2). Notice that the adjoint variable $\overline{P}$ in (3.3.10), with its terminal condition, is obtained by marching backward in time. As a result, the equation (3.3.10) is called *backward propagation.*

(C) **Design Equations.** Finally, equating $\nabla_W \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P})$, $\nabla_K \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P})$, and $\nabla_b \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \overline{b}; \overline{P})$ to zero, respectively, yields the design equations (with $(\overline{W}, \overline{K}, \overline{b})$ as the design variables),

$$
\begin{aligned}
\nabla_W \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& \frac{1}{n}\big(- C_{obs} + S(\overline{W}, \overline{Y}(T))\big) \, \big(\overline{Y}(T)\big)^{\intercal} \\
& + \nabla_W \mathcal{R}(\overline{W}, \overline{K}(T), \bar{b}(T)) = 0, \\
\nabla_K \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& - \overline{Y}(t) \, \Big(\overline{P}(t) \odot \sigma'(\overline{K}(t)\overline{Y}(t) + \bar{b}(t))\Big)^{\intercal} \\
& + \nabla_K \mathcal{R}(\overline{W}, \overline{K}(t), \bar{b}(t)) = 0, \\
\nabla_b \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& - \langle \sigma'(\overline{K}(t)\overline{Y}(t) + \bar{b}(t)), \overline{P}(t) \rangle_F \\
& + \nabla_b \mathcal{R}(\overline{W}, \overline{K}(t), \bar{b}(t)) = 0,
\end{aligned}
\tag{3.3.11}
$$

for almost every $t \in (0, T)$.

In view of (A)-(C), we can use a gradient based solver to find a stationary point to (3.3.7).

Remark 3.3.2. (Parametric Kernel $K(\psi(t)))$. Throughout our discussion, we have assumed $K(t)$ to be some unknown linear operator. We remark that a structure could also be prescribed to $K(t)$, parameterized by a stencil ψ. Then, the kernel is $K(\psi(t))$, and the design variables now are $\theta = (W, \psi, b)$. Consequently, $K(\psi(t))$ can be thought of as a differential operator on the feature space, e.g. discrete Laplacian with a five point stencil. It then remains to compute the sensitivity of the Lagrangian functional w.r.t. ψ to get the design equation. Note that this approach can further reduce the number of unknowns. □

Notice that so far the entire discussion has been at the continuous level and it has been independent of the number of network layers. Thus, it is expected that if we discretize (in time) the above optimality system, then the resulting gradient based solver is independent of the number of layers. We shall discretize the above optimality system in the next section.

3.4 Discrete Fractional Deep Neural Network

We shall adopt the *optimize-then-discretize* approach. Recall that the first order stationarity conditions for the continuous problem (3.3.7) are given in (3.3.9), (3.3.10), and (3.3.11). In order to discretize this system of equations, we shall first discuss the approximation of Caputo fractional derivative.

3.4.1 Approximation of Caputo Derivative

There exist various approaches to discretize the fractional Caputo derivative. We will use the L^1-scheme [15, 136] to discretize the left and right Caputo fractional derivative $d_t^\gamma u(t)$ and $d_{T-t}^\gamma u(t)$ given in (3.2.3) and (3.2.4), respectively. Exploration of other discretizations would be part of our future work.

Consider the following fractional differential equation involving the **left** Caputo fractional derivative, for $0 < \gamma < 1$,

$$d_t^\gamma u(t) = f(u(t)), \quad u(0) = u_0. \tag{3.4.1}$$

We begin by discretizing the time interval $[0, T]$ uniformly with step size τ,

$$0 = t_0 < t_1 < t_2 < \cdots < t_{j+1} < \cdots < t_N = T, \text{ where } t_j = j\tau.$$

Then using the L^1-scheme, the discretization of (3.4.1) is given by

$$u(t_{j+1}) = u(t_j) - \sum_{k=0}^{j-1} a_{j-k} \, \big(u(t_{k+1}) - u(t_k)\big) + \tau^\gamma \Gamma(2-\gamma) f(u(t_j)). \quad j = 0, ..., N-1 \ , \tag{3.4.2}$$

where coefficients a_k are given by,

$$a_{j-k} = (j+1-k)^{1-\gamma} - (j-k)^{1-\gamma}. \tag{3.4.3}$$

Next, let us consider the discretization of the fractional differential equation involving the **right** Caputo fractional operator, for $0 < \gamma < 1$,

$$d_{T-t}^{\gamma} u(t) = f(u(t)), \quad u(T) = u_T. \tag{3.4.4}$$

Again using L^1-scheme we get the following discretization of (3.4.4):

$$u(t_{j-1}) = u(t_j) + \sum_{k=j}^{N-1} a_{k-j} \, \big(u(t_{k+1}) - u(t_k)\big) - \tau^{\gamma}\Gamma(2-\gamma) f(u(t_j)). \quad j = N, ..., 1. \tag{3.4.5}$$

The example below illustrates a numerical implementation of the L^1-scheme (3.4.2).

Example 3.4.1. Consider the linear differential equation

$$d_t^{0.5} u(t) = -4u(t), \;\; u(0) = 0.5. \tag{3.4.6}$$

Then, the solution to (3.4.6) is given by, see [125, Section 42], also [116, Section 1.2]

$$u(t) = 0.5 \, E_{0.5}(-4t^{0.5}), \tag{3.4.7}$$

where E_α, with $\alpha > 0$, is the Mittag Leffler function defined by

$$E_\alpha(z) = E_{\alpha,1}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}.$$

Figure 3.1 depicts the true solution and the numerical solutions using discretization (3.4.2) for the above example with uniform step size $\tau = 0.005$ and final time, $T = 1$. □

3.4.2 Discrete Optimality Conditions

Next, we shall discretize the optimality conditions given in (3.3.9) – (3.3.11). Notice that, each time-step corresponds to one layer of the neural network. It is necessary to do one

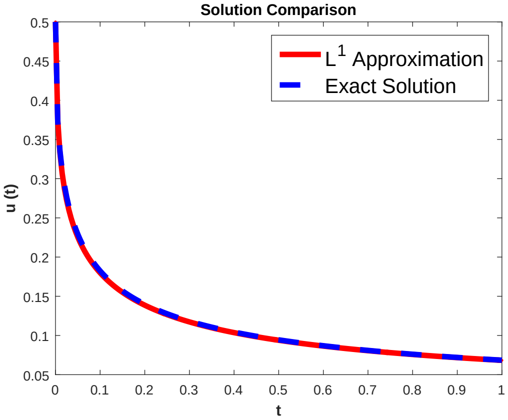

Figure 3.1: Comparison of the exact solution of (3.4.6) (*blue*) with an L^1 scheme approximation (*red*).

forward propagation (state solve) and one backward propagation (adjoint solve) to derive an expression of the gradient with respect to the design variables.

(A) **Discrete State Equation.** We use the L^1 scheme discussed in (3.4.2) to discretize the state equation (3.3.9) and arrive at

$$\begin{aligned}\overline{Y}(t_j) &= Y(t_{j-1}) - \sum_{k=1}^{j-1} a_{j-k}\, \big(Y(t_k) - Y(t_{k-1})\big) \\ &\qquad + \tau^{\gamma}\Gamma(2-\gamma)\sigma(\overline{K}(t_{j-1})\overline{Y}(t_{j-1}) + \overline{b}(t_{j-1})), \qquad j = 1, ..., N \\ \overline{Y}(t_0) &= Y_0\end{aligned} \tag{3.4.8}$$

(B) **Discrete Adjoint Equation.** We use the L^1 scheme discussed in (3.4.5) to discretize the adjoint equation (3.3.10) and arrive at

$$
\begin{aligned}
\overline{P}(t_j) &= P(t_{j+1}) + \sum_{k=j+1}^{N-1} a_{k-j-1}\, \big(P(t_{k+1}) - P(t_k)\big) - \qquad\qquad j = N-1, ..., 0 \\
&\qquad \tau^{\gamma}\Gamma(2-\gamma)\left[-\overline{K}(t_j)^{\intercal}\left(\overline{P}(t_{j+1}) \odot \sigma'\left(\overline{K}(t_j)\overline{Y}(t_{j+1}) + \bar{b}(t_j)\right)\right)\right], \\
\overline{P}(t_N) &= -\frac{1}{n}\overline{W}^{\intercal}(-C_{obs} + S(\overline{W}, \overline{Y}(t_N)))
\end{aligned}
\tag{3.4.9}
$$

(C) **Discrete Gradient w.r.t. Design Variables.** For $j = 0, \ldots, N-1$, the approximation of the gradient (3.3.11) with respect to the design variables is given by,

$$
\begin{aligned}
\nabla_W \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& \frac{1}{n}\big(- C_{obs} + S(\overline{W}, \overline{Y}(t_N))\big)\ \big(\overline{Y}(t_N)\big)^{\intercal} \\
&+ \nabla_W \mathcal{R}(\overline{W}, \overline{K}(t_N), \bar{b}(t_N)) \\
\nabla_K \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& - \overline{Y}(t_j)\ \left(\overline{P}(t_{j+1}) \odot \sigma'(\overline{K}(t_j)\overline{Y}(t_j) + \bar{b}(t_j))\right)^{\intercal} \\
&+ \nabla_K \mathcal{R}(\overline{W}, \overline{K}(t_j), \bar{b}(t_j)) \\
\nabla_b \mathcal{L}(\overline{Y}, \overline{W}, \overline{K}, \bar{b}; \overline{P}) =& -\ \langle \sigma'(\overline{K}(t_j)\overline{Y}(t_j) + \bar{b}(t_j)), \overline{P}(t_{j+1})\rangle_F \\
&+ \nabla_b \mathcal{R}(\overline{W}, \overline{K}(t_j), \bar{b}(t_j))\ .
\end{aligned}
\tag{3.4.10}
$$

Whence, we shall create a gradient based method to solve the optimality condition (3.4.8)-(3.4.10). We reiterate that each computation of the gradient in (3.4.10), requires one state and one adjoint solve.

3.5 Fractional-DNN Algorithm

Fractional-DNN is a supervised learning architecture, i.e. it comprises of a training phase and a testing phase. During the training phase, labeled data is passed into the network and the unknown parameters are learnt. Those parameters then define the trained Fractional-DNN model for that type of data. Next, a testing dataset, which comprises of data previously unseen by the network, is passed to the trained net, and a prediction of classification is obtained. This stage is known as the testing phase. Here the true classification is not shown to the network when a prediction is being made, but can later be used to compare the network efficiency, as we have done in our numerics. The three important components of the algorithmic structure are forward propagation, backward propagation, and gradient update. The forward and backward propagation structures are given in Algorithms 3 and 4. The gradient update is accomplished in the training phase, discussed in Subsection 3.5.1. Lastly, the testing phase of the algorithm is discussed in Subsection 3.5.2.

Algorithm 3 Forward Propagation in Factional-DNN (L^1-scheme)

```
Input: $(Y_0, C_{obs}), W, \{K_j, b_j\}_{j=0}^{N-1}, N, \tau, \gamma$
Output: $\{Y_j\}_{j=1}^{N}, P_N,$
Let $z_0 = 0$.
for $j = 1, \cdots, N$ do
    for $k = 1, \cdots, j-1$ do
        Compute $a_{j-k}$:    {Use (3.4.3)}
        Update $z_k$:         $z_k = z_{k-1} + a_{j-k}(Y_k - Y_{k-1})$
    end for
    Update $Y_j$:          $Y_j = Y_{j-1} - z_{j-1} + (\tau)^\gamma \Gamma(2-\gamma) \sigma(K_{j-1}Y_{j-1} + b_{j-1})$
end for
Compute $P_N$:          $P_N = -(n)^{-1} W^\intercal(-C_{obs} + S(W, Y_N))$
```

3.5.1 Training Phase

The training phase of Fractional-DNN is shown in Algorithm 5.

3.5.2 Testing Phase

The testing phase of Fractional-DNN is shown in Algorithm 6.

Algorithm 4 Backward Propagation in Factional-DNN (L^1-scheme)

```
Input: $\{Y_j\}_{j=1}^{N},\ P_N,\ \{K_j, b_j\}_{j=0}^{N-1},\ N,\ \tau,\ \gamma$
Output: $\{P_j\}_{j=0}^{N-1}$
Let $x_0 = 0$.
for $j = N-1, \cdots, 0$ do
    for $k = j+1, \cdots, N-1$ do
        Compute $a_{k-j-1}$:    {Use (3.4.3)}
        Compute $x_k$:    $x_k = x_{k-1} + a_{k-j-1}\,(P_{k+1} - P_k)$
    end for
    Update $P_j$:    $P_j = P_{j+1} + x_{N-1} - (\tau)^{\gamma}\,\Gamma(2-\gamma)\,[-K_j^{\intercal}(P_{j+1} \odot \sigma'(K_j Y_{j+1} + b_j))]$
end for
```

Algorithm 5 Training Phase of Factional-DNN

```
Input: $(Y_0, C_{obs})\,,\ N,\ \tau,\ \gamma,\ m_1, m_2$
Output: $W,\ \{K_j, b_j\}_{j=0}^{N-1},\ C_{train}, \alpha_{train},$
Initialize $W, \{K_j, b_j\}_{j=0}^{N-1}$
for $i = 1, \cdots, m_1$ do
    Let $(\hat{Y}_0, \hat{C}_{obs}) \subset (Y_0, C_{obs})$ {Randomly select a mini-batch and apply BN using (3.6.1)}
    FORWARD PROPAGATION {Use Algorithm 3 to get $\{\hat{Y}_j\}_{j=1}^{N},\ P_N$}.
    BACKWARD PROPAGATION {Use Algorithm 4 to get $\{P_j\}_{j=0}^{N-1}$}.
    GRADIENT COMPUTATION
    Compute $\nabla_W \mathcal{L},\ \{\nabla_K \mathcal{L}\},\ \{\nabla_b \mathcal{L}\}$

        $$\begin{aligned}
        \nabla_W \mathcal{L} &= (n)^{-1}\left(-C_{obs} + S(W, \hat{Y}_N)\right)\ (\hat{Y}_N)^{\intercal} + \nabla_W \mathcal{R}(W, K_j, b_j) \\
        \nabla_K \mathcal{L} &= -\,\hat{Y}_j\ \left(P_{j+1} \odot \sigma'(K_j \hat{Y}_j + b_j)\right)^{\intercal} + \nabla_K \mathcal{R}(W, K_j, b_j) \\
        \nabla_b \mathcal{L} &= -\,tr\left(\sigma'(K_j \hat{Y}_j + b_j)\ P_{j+1}\right) + \nabla_b \mathcal{R}(W, K_j, b_j)
        \end{aligned}$$

    Pass $\nabla_W \mathcal{L},\ \nabla_K \mathcal{L},\ \nabla_b \mathcal{L}$ to gradient based solver with $m_2$ max iterations to update
    $W, \{K_j, b_j\}_{j=0}^{N-1}$.
    Compute $\hat{C}_{train} = S(W, \hat{Y}_N)$
    Compare $\hat{C}_{train}$ to $\hat{C}_{obs}$ to compute $\alpha_{train}$
end for
```

Algorithm 6 Testing Phase of Fractional-DNN

```
Input: $\left(Y_0^{test}, C_{obs,test}\right)$, $W$, $\{K_j, b_j\}_{j=0}^{N-1}$, $N$, $\tau$, $\gamma$
Output: $C_{test}, \alpha_{test}$
1: Let $Y_0 = Y_0^{test}$ {Apply BN using (3.6.1)}
2: FORWARD PROPAGATION {Use Algorithm 3 to get $\{Y_j\}_{j=1}^{N}$}.
3: Compute $C_{test} = S(W, Y_N)$
4: Compare $C_{test}$ to $C_{obs,test}$ to compute $\alpha_{test}$
```

3.6 Numerical Experiments

In this section, we present several numerical experiments where we use our proposed Fractional-DNN algorithm from Section 3.5 to solve classification problems for two different datasets. We recall that the goal of classification problems, as the name suggests, is to classify objects into pre-defined class labels. First we prepare a training dataset and along-with its classification, pass it to the training phase of Fractional-DNN (Algorithm 5). This phase yields the optimal set of parameters learned from the *training dataset*. They are then used to classify new data points from the *testing dataset* during the testing phase of Factional-DNN (Algorithm 6). We compare the results of our Fractional-DNN with the classical RNN (3.3.4).

The rest of this section is organized as follows: First, we discuss some data preprocessing and implementation details. Then we describe the datasets being used, and finally we present the experimental results.

3.6.1 Implementation Details

(i) **Batch Normalization.** During the training phase, we use the batch normalization (BN) technique [90]. At each iteration we randomly select a mini-batch, which comprises of 50% of the training data. We then normalize the mini-batch $\hat{Y}_0 \subset Y_0$, to have a zero mean and a standard deviation of one, i.e.

$$\hat{Y}_0 = \frac{\hat{Y}_0 - \mu(\hat{Y}_0)}{s(\hat{Y}_0)}, \tag{3.6.1}$$

where μ is the mean and s is the standard deviation of the mini-batch. The normalized mini-batch is then used to train the network in that iteration. At the next iteration, a new mini-batch is randomly selected. This process is repeated m_2 times. Batch normalization prevents gradient blow-up, helps speed up the learning and reduces the variation in parameters being learned.

Since the design variables are learnt on training data processed with BN, we also process the testing data with BN, in which case the mini-batch is the whole testing data.

(ii) **Activation Function.** For the experiments we have performed, we have used the hyperbolic tangent function as the activation function, for which,

$$\sigma(x) = \tanh(x), \quad \text{and} \quad \sigma'(x) = 1 - \tanh^2(x).$$

(iii) **Regularization.** In our experiments, we have used the following regularization:

$$\mathcal{R}(W, K, b) := \frac{\xi_W}{2}\|W\|_F^2 + \frac{\xi_K}{2N}\|(-\Delta)_h K(t)\|_F^2 + \frac{\xi_b}{2N}\|b(t)\|_2^2$$

where $(-\Delta)_h$ is the discrete Laplacian, and ξ_W, ξ_K, ξ_b are the scalar regularization strengths, and $\|\cdot\|_F$ is the Frobenius norm.

Notice that with the above regularization, we are enforcing Laplacian-smoothing on K. For a more controlled smoothness, one could also use the fractional Laplacian regularization introduced in [6], see also [18] and [8].

(iv) **Order of Fractional Time Derivative.** In our computations, we have chosen γ heuristically. We remark that this fractional exponent on time derivative can be learnt in a similar manner as the fractional exponent on Laplacian was learnt in [8], or as the authors did in [115].

(v) **Optimization Solver and Xavier Initialization.** The optimization algorithm we have used is the BFGS method with Armijo line search [93]. The stopping tolerance for the BFGS algorithm is set to $1e-6$ or maximum number of optimization iterations m_2, whichever is achieved first. However, in our experiments, the latter is achieved first in most cases. The design variables are initialized using Xavier initialization [67], according to which, the biases b are initialized as 0, and the entries of W, and K_j are drawn from the uniform distribution $U[-a, a]$. We consider $a = \sqrt{\frac{3}{n_f}}$ for the activation function $\sigma(\cdot) = \tanh(\cdot)$, and $a = \frac{1}{\sqrt{n_f}}$ for other activation functions.

(vi) **Network Layers vs. the Final Time.** For our experiments, we heuristically choose the number of layers N, and the discretization step-length for forward and backward propagation as $\tau = 0.2$. Thus our final time is given by, $T = t_N = N\tau$.

(vii) **Classification Accuracy.** We remark that when we calculate $C_{train} = S(W, Y_N)$, we obtain a probability distribution of the samples belonging to the classes. We consider the class with the highest probability as the predicted class. Then, we use a very standard procedure to compare C_{train} with C_{obs}.

$$\mathrm{n}_{\mathrm{cor,train}} := \text{No. of correctly identified labels} = n - \frac{1}{2}\|C_{obs} - C_{train}\|_F^2.$$

$$\text{training error} = 1 - \frac{\mathrm{n}_{\mathrm{cor,train}}}{n}, \quad \text{and} \quad \alpha_{train} = \frac{\mathrm{n}_{\mathrm{cor,train}}}{n} \times 100.$$

The same procedure is used to compute C_{test} and α_{test}.

(viii) **Gradient Test.** To verify the gradients in (3.4.10), we perform a gradient test by comparing them to a finite difference gradient approximation of (3.3.7). In Figure 3.2 we show that the two conform and we obtain the expected order of convergence for all the design variables.

(ix) **Computational Platform.** All the computations have been carried out in MATLAB R2015b on a laptop with an Intel Core i7-8550U processor.

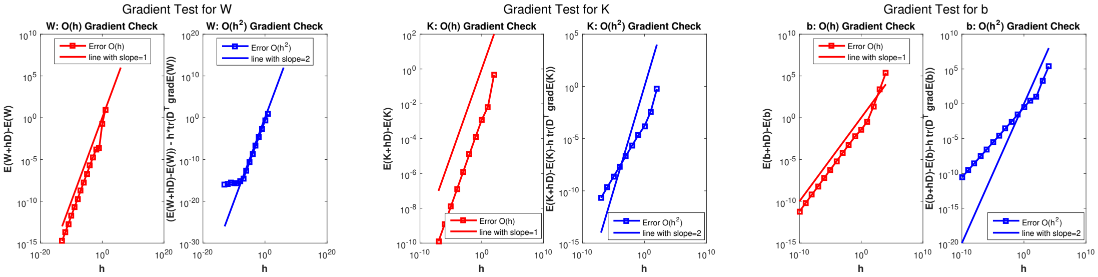

Figure 3.2: Comparison between derivative with respect to the design variables and finite difference approximation. The expected rate of convergence is obtained.

3.6.2 Experimental Datasets

We describe the datasets we have used to validate our proposed Fractional-DNN algorithm below.

- **Dataset 1: Coordinate to Level Set (CLS).** This data comprises of a set of 2D coordinates, i.e. $Y_0 := \{(x_i, y_i) \ | \ i = 1, \cdots, n; \ (x_i, y_i) \in \mathbb{R}^2([0,1])\}$. Next, we consider the following piecewise function,

$$v(x,y) = \begin{cases} 1 & \forall\, x \le y \\ \\ 0 & \forall\, x > y \end{cases} \qquad \forall \ \ x, y \in [0,1]. \tag{3.6.2}$$

  The coordinates are the features in this case, hence $n_f = 2$. Further, we have $n_c = 2$ classes, which are the two level sets of $v(x,y)$. Thus, for the ith sample $Y_0^{(i)}$, $C_{obs}^{(i)} \in \mathbb{R}^{n_c}$ is a standard basis vector which represents the probability that $Y_0^{(i)}$ belongs to the class label $\{1, 2\}$.

- **Dataset 2: Perfume Data (PD) [60, 62].** This dataset comprises of odors of 20 different perfumes measured via a handheld meter (OMX-GR sensor) every second, for 28 seconds. For this data, $Y_0 := \{(x_i, y_i) \mid i = 1, \cdots, n;\ x_i, y_i \in \mathbb{Z}_+\}$, thus $n_f = 2$. The classes, $n_c = 20$, pertain to 20 different perfumes. we construct C_{obs} in the same manner as we did for Dataset 1.

3.6.3 Forward Propagation as a Dynamical System

In the introduction we mentioned the idea of representing a DNN as an optimization problem constrained by a dynamical system. This has turned out to be a strong tool in studying the underlying mathematics of DNNs. In Figure 3.3 we numerically demonstrate how this viewpoint enables a more efficient strategy for distinguishing between the classes. First we consider the perfume data, which has two features, namely the (x, y) coordinates, and let it flow, i.e. forward propagate. When this evolved data is presented to the classifier functional (e.g. softmax function in our case), a spatially well-separated data is easier to classify. We plot the input data Y_0, represented as squares, as well as the evolved data Y_N after it has passed through N layers. The 20 different colors correspond to the 20 different classes for the data, which help us visually track the evolution from Y_0 to Y_N. The evolution under standard RNN is shown in the left plot, and that of Fractional-DNN is shown in the right plot. The configuration for these plots is the same as discussed in Subsection 3.6.5 below and pertains to the trained models. Notice that at the bottom right corner of the RNN evolution plot, the purple, pink and red data points are overlapping which poses a challenge for the classifier to distinguish between the classes. In contrast, Fractional-DNN has separated out those points quite well.

We remark that this separation also gives a hint as to the number of layers needed in a network. We need enough number of layers which would let the data evolve enough to be easily separable. However, the visualization can get restricted to $n_f \leq 3$, therefore for data with $n_f > 3$, it may be challenging to get a sense of number of layers needed to make the data separable-enough.

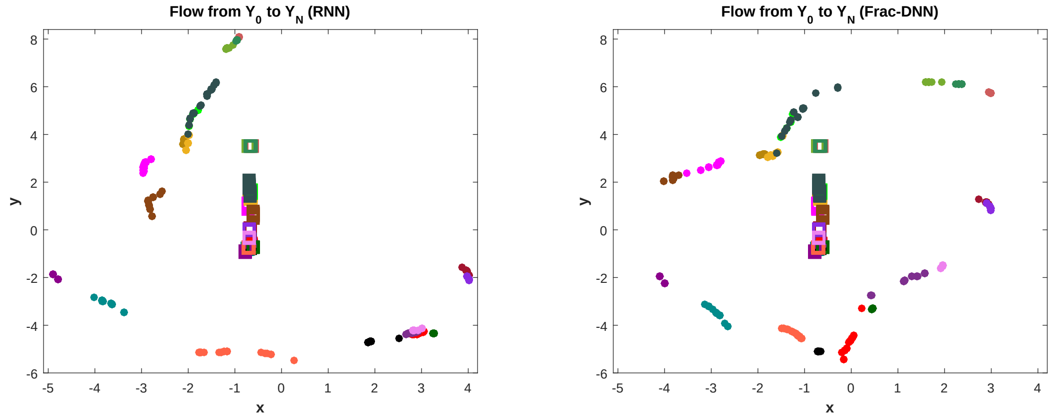

Figure 3.3: Forward propagation of perfume data from Y_0 (*squares*) to Y_N (*dots*) via standard RNN (*left*) and Fractional-DNN (*right*). Note that data is more linearly separable for Fractional-DNN. Different colors represent different classes

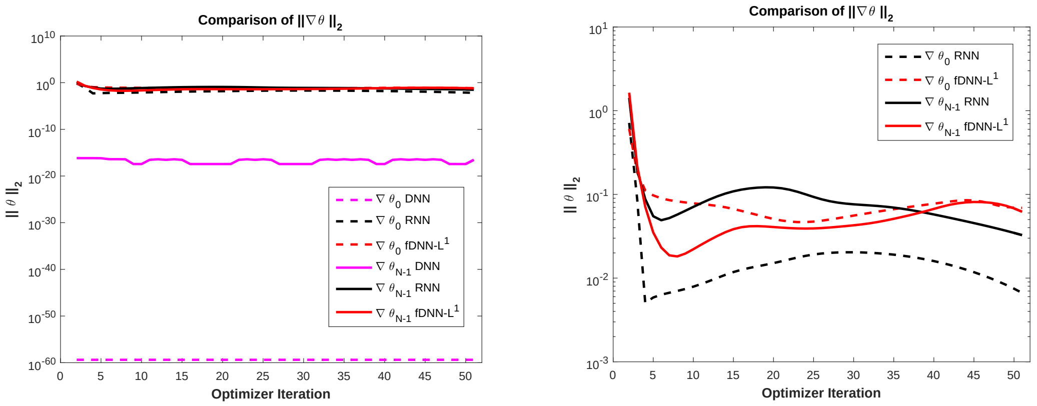

Figure 3.4: Demonstration of the gradient norm of $\theta = (K, b)$ at the first layer(dotted line) and last layer (solid line) of the network for various algorithms, namely standard DNN (magenta), RNN (black), and Fractional-DNN with L^1 scheme approximation (red). The figure on the *right* is the zoomed in version of figure on the *left*. Note the improvement in relative gradient propagation across layers for Fractional-DNN which leads to better learning ability and improves the vanishing gradient issue.

3.6.4 Vanishing Gradient Issue

In earlier sections, we remarked that Fractional-DNN handles the vanishing gradient issue in a better way. The vanishing gradient issue arises when the gradient of the design variables vanishes across the layers as the network undergoes backpropagation, see [69] and references therein. As a consequence, feature extraction in the initial layers gets severely affected, which in turn affects the learning ability of the whole network. We illustrate this phenomenon for the networks under discussion in Figure 3.4. In the *left* plot of Figure 3.4, we compare the $\| \cdot \|_2$ of the gradient of design variables $\theta = (K, b)$ against optimization solver (steepest descent in this case) iterations for standard DNN (which does not have any skip connection) in *magenta*, classical RNN (3.3.4) in *black*, and Fractional-DNN with L^1-scheme approximation from Algorithm 5 in red. In the *right* plot of Figure 3.4 we have omitted the standard DNN plot to take a closer look at the other two. Observe that as gradient information propagates backward, i.e. from layer $N-1$ to 0, its magnitude reduces by one order in the case of standard RNN. This implies that enough information is not being passed to initial layers relative to the last layer. In contrast, the Fractional-DNN is carrying significant information back to the initial layers while maintaining a relative magnitude. This improves the overall health of the network and improves learning. This test has been performed on Perfume Data (Dataset 2) with 70 layers and regularization turned off.

3.6.5 Experimental Results

We now solve the classification problem (3.3.7) for the datasets described in Subsection 3.6.2 via our proposed Fractional-DNN algorithm, presented in Section 3.5. We then compare it with the standard RNN architecture (3.3.4). The details and results of our experiments are given in Table 3.2.

Note that the results obtained via Fractional-DNN are either comparable to (e.g. for CLS data) or significantly better than (e.g. for PD) the standard RNN architecture.

We remark that while CLS data (Dataset 1) is a relatively simpler problem to solve

Table 3.2: Comparison of classification accuracy for various datasets using the standard RNN (3.3.4) with our proposed Fractional-DNN (3.3.6) with L^1 scheme approximation. Note the improvement in results due to Fractional-DNN.

Dataset	**CLS**	**CLS**	**PD**	**PD**
Time Derivative	Standard	Frac-L^1	Standard	Frac-L^1
$\boldsymbol{\gamma}$	–	0.1	–	0.9
$\boldsymbol{n_{train}}$	10000	10000	560	560
$\boldsymbol{n_{test}}$	10000	10000	532	532
N	5	5	35	35
$\boldsymbol{m_1}$	6	6	567	567
$\boldsymbol{m_2}$	30	30	15	15
$\boldsymbol{\xi_W}$	$1e-1$	$1e-1$	$1e-8$	$1e-8$
$\boldsymbol{\xi_K}$	$1e+2$	$1e+2$	0	0
$\boldsymbol{\xi_b}$	$1e-2$	$1e-2$	0	0
$\boldsymbol{\alpha_{train}}$	99.76%	99.82%	52.86%	70.36%
$\boldsymbol{\alpha_{test}}$	99.79%	99.79%	45.49%	84.21%

(two features and two classes), the Perfume Data (Dataset 2) is not. In the latter case, each dataset comprises of only two features, and there are 20 different classes. Furthermore, the number of available samples for training is small. In this sense, classification of this dataset is a challenging problem. There have been some results on classification of perfume data using only the training dataset (divided between training and testing) [62], but to the best of our knowledge, classification on the complete dataset using both the training and testing sets [60] is not available.

In our experiments, we have also observed that Fractional-DNN algorithm needs lesser number of Armijo line-search iterations than the standard RNN. This directly reflects an improvement in the learning rate via Fractional-DNN. We remark that in theory, Fractional-DNN should use memory more efficiently than other networks, as it encourages feature reuse in the network.

3.7 Discussion

There is a growing body of research which indicates that deep learning algorithms, e.g. a residual neural network, can be cast as optimization problems constrained by ODEs or PDEs. In addition, thinking of continuous optimization problems can make the approaches machine/architecture independent. This opens a plethora of tools from constrained optimization theory which can be used to study, analyze, and enhance the deep learning algorithms. Currently, the mathematical foundations of many machine learning models are largely lacking. Their success is mostly attributed to empirical evidence. Hence, due to the lack of mathematical foundation, it becomes challenging to fix issues, like network instability, vanishing and exploding gradients, long training times, inability to approximate non-smooth functions, etc., when a network breaks down.

In this work we have developed a novel continuous model and stable discretization of deep neural networks that incorporate history. In particular, we have developed a fractional deep neural network (Fractional-DNN) which allows the network to admit memory across all the subsequent layers. We have established this via an optimal control problem formulation of a deep neural network bestowed with a fractional time Caputo derivative. We have then derived the optimality conditions using the Lagrangian formulation. We have also discussed discretization of the fractional time Caputo derivative using L^1-scheme and presented the algorithmic framework for the discretization.

We expect that by keeping track of history in this manner improves the vanishing gradient problem and can potentially strengthen feature propagation, encourage feature reuse and reduce the number of unknown parameters. We have numerically illustrated the improvement in the vanishing gradient issue via our proposed Fractional-DNN. We have shown that Fractional-DNN is better capable of passing information across the network layers which maintains the relative gradient magnitude across the layers, compared to the standard DNN and standard RNN. This allows for a more meaningful feature extraction to happen at each layer.

We have shown successful application of Fractional-DNN for classification problems using various datasets, namely the Coordinate to Level Set and Perfume Data. We have compared the results against the standard-RNN and have shown that the Fractional-DNN algorithm yields improved results.

We emphasis that our proposed Fractional-DNN architecture has a *memory* effect due to the fact that it allows propagation of features in a cumulative manner, i.e. at each layer all the precedent layers are visible. Reusing the network features in this manner reduces the number of parameters that the network needs to learns in each subsequent layer. Fractional-DNN has a rigorous mathematical foundation and algorithmic framework which establishes a deeper understanding of deep neural networks with memory. This enhances their applicability to scientific and engineering applications.

We remark that code optimization is part of our forthcoming work. This would involve efficient Graphic Processing Unit usage and parallel computing capabilities. We intend to develop a python version of this code and incorporate it into popular deep learning libraries like TensorFlow, PyTorch etc. We are also interested in expanding the efficiency of this algorithm to large-scale problems suitable for High Performance Computing.

Chapter 4: External Optimal Control of Nonlocal PDEs [1]

4.1 Introduction and Motivation

In many real life applications a source or a control is placed outside (disjoint from) the observation domain Ω where the PDE is satisfied. Some examples of inverse and optimal control problems where this situation may arise are: (i) Acoustic testing, when the loudspeakers are placed far from the aerospace structures [98]; (ii) Magnetotellurics (MT), which is a technique to infer earth's subsurface electrical conductivity from surface measurements [140,149]; (iii) Magnetic drug targeting (MDT), where drugs with ferromagnetic particles in suspension are injected into the body and the external magnetic field is then used to steer the drug to relevant areas, for example, solid tumors [13,14,105]; (iv) Electroencephalography (EEG) is used to record electrical activities in brain [112,151], in case one accounts for the neurons disjoint from the brain, one will obtain an external source problem.

This is different from the traditional approaches where the source/control is placed either inside the domain Ω or on the boundary $\partial\Omega$ of Ω. This is not surprising since in many cases we do not have a direct access to $\partial\Omega$. See for instance, the setup in Figure 4.1. In such applications the existing models can be ineffective due to their strict requirements. Indeed think of the source identification problem for the most basic Poisson equation:

$$-\Delta u = f \quad \text{in } \Omega, \quad u = z \quad \text{on } \partial\Omega, \tag{4.1.1}$$

where the source is either f (force or load) or z (boundary control) see [11,101,139]. In (4.1.1) there is no provision to place the source in $\widehat{\Omega} \subset \mathbb{R}^N \setminus \Omega$, i.e., a domain that is disjoint from Ω, see Figure 4.1 for two examples of Ω and $\widehat{\Omega}$. The issue is that the operator Δ

[1] This work has been published in [10].

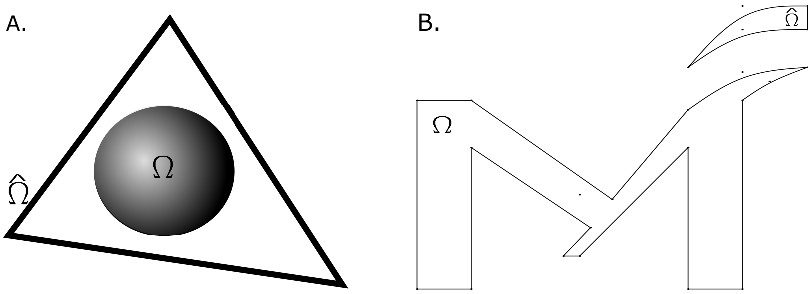

Figure 4.1: Let a diffusion process occurs inside a domain Ω which is the sphere in Case. A (left) and the letter M in Case. B (right). We are interested in the source identification or controlling this diffusion process by placing the source/control in a set $\widehat{\Omega}$ which is disjoint from Ω. Case A: $\widehat{\Omega}$ is the triangular pipe. Case B: $\widehat{\Omega}$ is the structure on the top of the letter M.

has "lesser reach", in other words, it is a local operator. On the other hand the fractional Laplacian $(-\Delta)^s$ with $0 < s < 1$ (that we shall define below) is a nonlocal operator. This difference in behavior can be easily seen in our numerical examples in Subsection 4.7.2 where we observe that we cannot see the external source as s approaches 1.

Recently, nonlocal diffusion operators such as the fractional Laplacian $(-\Delta)^s$ have emerged as an excellent alternative to model diffusion. Under a probabilistic framework this operator can be derived as the limit of the so-called *long jump* random walk [141]. Recall that Δ is the limit of the classical random walk or the Brownian motion. More applications of these models appear in (but not limited to) image denoising and phase field modeling [6,18]; fractional diffusion maps (data analysis) [7]; magnetotellurics (geophysics) [149].

Coming back to the question of source/control placement, we next state the exterior value problem corresponding to $(-\Delta)^s$. Find u in an appropriate function space satisfying

$$(-\Delta)^s u = f \quad \text{in } \Omega, \quad u = z \quad \text{on } \mathbb{R}^N \setminus \Omega. \tag{4.1.2}$$

As in the case of (4.1.1), besides f being the source/control in Ω, we can also place the source/control z in the exterior domain $\mathbb{R}^N \setminus \Omega$. However, the action of z in (4.1.2) is significantly different from (4.1.1). Indeed, the source/control in (4.1.1) is placed on the boundary $\partial\Omega$, but the source/control z in (4.1.2) is placed in $\mathbb{R}^N \setminus \Omega$ which is what we wanted to achieve in Figure 4.1. For completeness, we refer to [21] for the optimal control problem, with f being the source/control and [20,22] for another inverse problem to identify the coefficients in the fractional p-Laplacian.

The purpose of this work is to introduce and study a new class of the Dirichlet, Robin and Neumann source identification problems or the optimal control problems. We shall use these terms interchangeably but we will make a distinction in our numerical experiments. We emphasize that yet another class of identification where the unknown is the fractional exponent s for the spectral fractional Laplacian (which is different from the operator under consideration) was recently considered in [132]. We refer to [18] for the case when s is a function of $x \in \Omega$.

Now we describe our problems.

Let $\Omega \subset \mathbb{R}^N$, $N \geq 1$, be a bounded open set with boundary $\partial\Omega$. Let (Z_D, U_D) and (Z_R, U_R), where the subscripts D and R indicate Dirichlet and Robin, be Banach spaces. The goal of this work is to consider the following two external control or source identification problems. The source/control in our case is denoted by z. Our objective function consists of two parts and we shall denote by J the part that depends on the state u. The precise assumptions on J will be given in Section 4.4.

- **Fractional Dirichlet Exterior Control Problem.** Given $\xi \geq 0$ a constant penalty parameter, we consider the minimization problem:

$$\min_{(u,z)\in(U_D,Z_D)} J(u) + \frac{\xi}{2}\|z\|^2_{Z_D}, \tag{4.1.3a}$$

subject to the fractional Dirichlet exterior value problem. Find $u \in U_D$ solving

$$\begin{cases} (-\Delta)^s u &= 0 \quad \text{in } \Omega, \\ u &= z \quad \text{in } \mathbb{R}^N \setminus \Omega, \end{cases} \tag{4.1.3b}$$

and the control constraints

$$z \in Z_{ad,D}, \tag{4.1.3c}$$

with $Z_{ad,D} \subset Z_D$ being a closed and convex subset.

- **Fractional Robin Exterior Control Problem.** Given $\xi \geq 0$ a constant penalty parameter, we consider the minimization problem:

$$\min_{(u,z)\in(U_R,Z_R)} J(u) + \frac{\xi}{2}\|z\|^2_{Z_R}, \tag{4.1.4a}$$

subject to the fractional Robin exterior value problem: Find $u \in U_R$ solving

$$\begin{cases} (-\Delta)^s u = 0 & \text{in } \Omega, \\ \mathcal{N}_s u + \kappa u = \kappa z & \text{in } \mathbb{R}^N \setminus \Omega, \end{cases} \tag{4.1.4b}$$

and the control constraints

$$z \in Z_{ad,R}, \tag{4.1.4c}$$

with $Z_{ad,R} \subset Z_R$ being a closed and convex subset. In (4.1.4b), $\mathcal{N}_s u$ is the nonlocal normal derivative of u that will be defined in Section 4.2, $\kappa \in L^1(\mathbb{R}^N \setminus \Omega) \cap L^\infty(\mathbb{R}^N \setminus \Omega)$ and is non-negative. We notice that the latter assumption is not a restriction since otherwise we can replace κ throughout by $|\kappa|$.

The precise conditions on Ω and the Banach spaces involved will be given in the subsequent sections. Notice that both exterior value problems (4.1.3b) and (4.1.4b) are ill-posed if the conditions are enforced on $\partial\Omega$. The main difficulties in (4.1.3) and (4.1.4) stem from the following facts.

- **Nonlocal Diffusion Operator.** The fractional Laplacian $(-\Delta)^s$ is a nonlocal operator. This can be easily seen from its definition.
- **Nonlocal Normal Derivative.** The first order optimality conditions for (4.1.3), the very-weak solution to the Dirichlet problem (4.1.3b) and the Robin exterior value problem (4.1.4b) require to study $\mathcal{N}_s u$ which is the so-called nonlocal-normal derivative of u. Thus we not only have the nonlocal operator $(-\Delta)^s$ in the domain but also in the exterior $\mathbb{R}^N \setminus \Omega$, i.e., a double nonlocality. An approximation of $\mathcal{N}_s u$, especially numerically, is extremely challenging.
- **Exterior Conditions in $\mathbb{R}^N \setminus \Omega$ and Not Boundary Conditions on $\partial\Omega$.** The conditions in (4.1.3b) and (4.1.4b) need to be specified in $\mathbb{R}^N \setminus \Omega$ instead on $\partial\Omega$ as otherwise the problems (4.1.3) and (4.1.4) are ill-posed as we have already mentioned above.
- **Very-Weak Solutions of Nonlocal Exterior Value Problems.** A typical choice for Z_D is $L^2(\mathbb{R}^N \setminus \Omega)$. As a result, the Dirichlet exterior value problem (4.1.3b) can only have very-weak solutions (cf. [23, 24, 32] for the case $s = 1$). To the best of our knowledge this is the first work that considers the notion of very-weak solutions for nonlocal (fractional) exterior value problems associated with the fractional Laplace operator.
- **Regularity of the Optimization Variables.** The standard shift-theorem which holds for local operators such as Δ does not always hold for nonlocal operators such as $(-\Delta)^s$ (see for example [73]).

In view of all these aforementioned challenges it is clear that the standard techniques which are now well established for local problems do not directly extend to the nonlocal problems investigated in this work.

The purpose of this work is to discuss our approach to deal with these nontrivial issues. We emphasize that to the best of our knowledge this is the first work that considers the optimal control problems (source identification problems) (4.1.3b) and (4.1.4b) where the control/source is applied from the outside. Let us also mention that this notion of controllability of PDEs from the exterior has been introduced by M. Warma in [147] for the nonlocal heat equation associated with the fractional Laplacian and in [103] for the wave type equation with the fractional Laplace operator to study their controllability properties. The case of the strong damping wave equation is included in [148] where some controllability results have been obtained. In case of problems with the spectral fractional Laplacian the boundary control has been established in [17]. For completeness, we also mention some interesting works on fractional Calderón type inverse problems [64, 97, 123]. Notice that fractional operators further provide flexibility to approximate arbitrary functions [46, 58, 73, 96].

We mention that we can also deal with the **fractional Neumann exterior control problem**. That is, given $\xi \geq 0$ a constant penalty parameter,

$$\min_{(u,z)\in(U_N,Z_N)} J(u) + \frac{\xi}{2}\|z\|^2_{Z_N},$$

subject to the fractional Neumann exterior value problem: Find $u \in U_N$ solving

$$\begin{cases} (-\Delta)^s u + u &= 0 \quad \text{in } \Omega, \\ \mathcal{N}_s u &= z \quad \text{in } \mathbb{R}^N \setminus \Omega, \end{cases} \tag{4.1.5}$$

and the control constraints

$$z \in Z_{ad,N}.$$

The term u is added in (4.1.5) just to ensure the uniqueness of solutions. The proofs follow similarly as the two cases we consider in this work with very minor changes. Since this Chapter is already long, we shall not give any details on this case.

Below we mention the novelties of this work.

(i) **Weak and Very-Weak Solutions.** For the first time, we introduce and study the notion of very-weak solutions to the Dirichlet exterior value problem (4.1.3b) which is suitable for optimal control problems. We also study weak solutions of the Robin exterior value problem (4.1.4b).

(ii) **Approximation of the Dirichlet Weak and Very-Weak Solutions by the Robin Weak Solutions.** We approximate the weak and very-weak solutions of the nonhomogeneous Dirichlet exterior value problem by using a suitable Robin exterior value problem. This allows us to circumvent approximating the nonlocal normal derivative and it is one of the *key contribution of this work.* Recall that for the very-weak solution of the Dirichlet problem we need to evaluate the nonlocal normal derivative of the test functions (see Theorem 4.3.3) and for the Dirichlet control problem we need to evaluate the nonlocal normal derivative of the adjoint variable (see Theorem 4.4.3). This is a new approach to impose non-zero exterior conditions for the fractional Dirichlet exterior value problem. We refer to an alternative approach [3] where the authors use the Lagrange multipliers to impose nonzero Dirichlet exterior conditions.

(iii) We study both Dirichlet and Robin exterior control problems.

(iv) We approximate (with rate) the Dirichlet exterior control problem by a suitable Robin exterior control problem.

The rest of the Chapter is organized as follows. We begin with Section 4.2 where we introduce the relevant notations and the function spaces needed. The material in this section is well-known. Our main work starts from Section 4.3 where we study first the weak and very-weak solutions for the Dirichlet exterior value problem in Subsection 4.3.1. This is followed by the well-posedness of the Robin exterior value problem in Subsection 4.3.2. The Dirichlet exterior control problem is considered in Section 4.4 and Robin in Section 4.5.

We show how to approximate the weak solutions to the Dirichlet problem and the solutions to the Dirichlet exterior control problem in Section 4.6. Subsection 4.7.1 is devoted to the experimental rate of convergence to approximate the Dirichlet exterior value problem using the Robin problem. In Subsection 4.7.2 we consider a source identification problem in the classical sense, however our source is located outside the observation domain where the PDE is satisfied. Subsection 4.7.3 is devoted to two optimal control problems.

Remark 4.1.1 (**Practical Aspects).** From a practical point of view, having the source/control over the entire $\mathbb{R}^N \setminus \Omega$ can be very expensive. But this can be easily fixed by appropriately describing Z_{ad}. Indeed in case of Figure 4.1 we can set the support of functions in Z_{ad} to be in $\widehat{\Omega}$.

4.2 Notations and Preliminaries

Unless otherwise stated, $\Omega \subset \mathbb{R}^N$ $(N \geq 1)$ is a bounded open set and $0 < s < 1$. We let

$$W^{s,2}(\Omega) := \left\{ u \in L^2(\Omega) \ : \ \int_\Omega \int_\Omega \frac{|u(x) - u(y)|^2}{|x-y|^{N+2s}} \, dxdy < \infty \right\},$$

and we endow it with the norm defined by

$$\|u\|_{W^{s,2}(\Omega)} := \left(\int_\Omega |u|^2 \, dx + \int_\Omega \int_\Omega \frac{|u(x) - u(y)|^2}{|x-y|^{N+2s}} \, dxdy \right)^{\frac{1}{2}}.$$

In order to study (4.1.3b) we also need to define

$$W_0^{s,2}(\overline{\Omega}) := \left\{ u \in W^{s,2}(\mathbb{R}^N) \ : \ u = 0 \text{ in } \mathbb{R}^N \setminus \Omega \right\}.$$

Then

$$\|u\|_{W_0^{s,2}(\overline{\Omega})} := \left(\int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}}\, dxdy \right)^{\frac{1}{2}}$$

defines an equivalent norm on $W_0^{s,2}(\overline{\Omega})$.

We shall use $W^{-s,2}(\mathbb{R}^N)$ and $W^{-s,2}(\overline{\Omega})$ to denote the dual spaces of $W^{s,2}(\mathbb{R}^N)$ and $W_0^{s,2}(\overline{\Omega})$, respectively, and $\langle \cdot, \cdot \rangle$ to denote their duality pairing whenever it is clear from the context.

We also define the local fractional order Sobolev space

$$W^{s,2}_{\mathrm{loc}}(\mathbb{R}^N \setminus \Omega) := \left\{ u \in L^2(\mathbb{R}^N \setminus \Omega) \ : \ u\varphi \in W^{s,2}(\mathbb{R}^N \setminus \Omega), \ \forall\, \varphi \in \mathcal{D}(\mathbb{R}^N \setminus \Omega) \right\}. \tag{4.2.1}$$

To introduce the fractional Laplace operator, we set

$$\mathbb{L}^1_s(\mathbb{R}^N) := \left\{ u : \mathbb{R}^N \to \mathbb{R} \ \text{ measurable}, \ \int_{\mathbb{R}^N} \frac{|u(x)|}{(1+|x|)^{N+2s}}\, dx < \infty \right\}.$$

For $u \in \mathbb{L}^1_s(\mathbb{R}^N)$ and $\varepsilon > 0$, we let

$$(-\Delta)^s_\varepsilon u(x) = C_{N,s} \int_{\{y \in \mathbb{R}^N, |y-x|>\varepsilon\}} \frac{u(x)-u(y)}{|x-y|^{N+2s}} dy, \ \ x \in \mathbb{R}^N,$$

where the normalized constant $C_{N,s}$ is given by

$$C_{N,s} := \frac{s 2^{2s} \Gamma\left(\frac{2s+N}{2}\right)}{\pi^{\frac{N}{2}} \Gamma(1-s)}, \tag{4.2.2}$$

and Γ is the usual Euler Gamma function (see, e.g. [36,42–44,56,145,146]). The **fractional Laplacian** $(-\Delta)^s$ is defined for $u \in \mathbb{L}^1_s(\mathbb{R}^N)$ by the formula

$$(-\Delta)^s u(x) = C_{N,s}\text{P.V.}\int_{\mathbb{R}^N} \frac{u(x)-u(y)}{|x-y|^{N+2s}}dy = \lim_{\varepsilon\downarrow 0}(-\Delta)^s_\varepsilon u(x), \quad x \in \mathbb{R}^N, \tag{4.2.3}$$

provided that the limit exists. It has been shown in [39, Proposition 2.2] that for $u \in \mathcal{D}(\Omega)$, we have

$$\lim_{s\uparrow 1^-}\int_{\mathbb{R}^N} u(-\Delta)^s u\, dx = \int_{\mathbb{R}^N} |\nabla u|^2 dx = -\int_{\mathbb{R}^N} u\Delta u\, dx = -\int_\Omega u\Delta u\, dx,$$

that is where the constant $C_{N,s}$ plays a crucial role.

Next, for $u \in W^{s,2}(\mathbb{R}^N)$ we define the nonlocal normal derivative $\mathcal{N}_s$ as:

$$\mathcal{N}_s u(x) := C_{N,s}\int_\Omega \frac{u(x)-u(y)}{|x-y|^{N+2s}}\, dy, \quad x \in \mathbb{R}^N \setminus \overline{\Omega}. \tag{4.2.4}$$

We shall call $\mathcal{N}_s$ the *interaction operator*. Notice that the term “interaction” has also been used by Du. et. al in [59]. Clearly $\mathcal{N}_s$ is a nonlocal operator and it is well defined on $W^{s,2}(\mathbb{R}^N)$ as we discuss next.

Lemma 4.2.1. The interaction operator $\mathcal{N}_s$ maps continuously $W^{s,2}(\mathbb{R}^N)$ into $W^{s,2}_{\text{loc}}(\mathbb{R}^N \setminus \Omega)$. As a result, if $u \in W^{s,2}(\mathbb{R}^N)$, then $\mathcal{N}_s u \in L^2(\mathbb{R}^N \setminus \Omega)$.

Proof. We refer to [65, Lemma 3.2] for the proof of the first part. The second part is a direct consequence of (4.2.1). □

Despite the fact that $\mathcal{N}_s$ is defined on $\mathbb{R}^N \setminus \Omega$, it is still known as the “normal” derivative. This is due to its similarity with the classical normal derivative as we discuss next.

Proposition 4.2.2. Let $\Omega \subset \mathbb{R}^N$ be a bounded open set with a Lipschitz continuous boundary. Then the following assertions hold.

(a) **The Divergence Theorem for $(-\Delta)^s$.** Let $u \in C_0^2(\mathbb{R}^N)$, i.e., C^2 functions on $\mathbb{R}^N$

that vanish at ±∞. Then

$$\int_{\Omega} (-\Delta)^s u \, dx = -\int_{\mathbb{R}^N \setminus \Omega} \mathcal{N}_s u \, dx.$$

(b) **The Integration by Parts Formula for $(-\Delta)^s$.** Let $u \in W^{s,2}(\mathbb{R}^N)$ be such that $(-\Delta)^s u \in L^2(\Omega)$. Then for every $v \in W^{s,2}(\mathbb{R}^N)$ we have that

$$\begin{aligned}\int_{\Omega} v(-\Delta)^s u \, dx =& \frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}} \, dxdy \\ & - \int_{\mathbb{R}^N \setminus \Omega} v \mathcal{N}_s u \, dx, \end{aligned} \tag{4.2.5}$$

where $\mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2 := (\Omega \times \Omega) \cup (\Omega \times (\mathbb{R}^N \setminus \Omega)) \cup ((\mathbb{R}^N \setminus \Omega) \times \Omega)$.

(c) **The limit as $s \uparrow 1^-$.** Let $u, v \in C_0^2(\mathbb{R}^N)$. Then

$$\lim_{s \uparrow 1^-} \int_{\mathbb{R}^N \setminus \Omega} v \mathcal{N}_s u \, dx = \int_{\partial\Omega} v \frac{\partial u}{\partial \nu} \, d\sigma.$$

Remark 4.2.3. Comparing the properties (a)-(c) in Theorem 4.2.2 with the classical properties of the standard Laplacian Δ we can immediately infer that $\mathcal{N}_s$ plays the same role for $(-\Delta)^s$ that the classical normal derivative does for Δ. For this reason, we call $\mathcal{N}_s$ the nonlocal normal derivative.

Proof of Theorem 4.2.2. The proofs of Parts (a) and (c) are contained in [57, Lemma 3.2] and [57, Proposition 5.1], respectively. The proof of Part (b) for smooth functions can be found in [57, Lemma 3.3]. The version given here is obtained by using a density argument (cf. [147, Proposition 3.7]). □

4.3 The State Equations

Before analyzing the optimal control problems (4.1.3) and (4.1.4) for a given function z, we shall focus on the Dirichlet (4.1.3b) and Robin (4.1.4b) exterior value problems. We shall assume that Ω is a bounded domain with a Lipschitz continuous boundary.

4.3.1 The Dirichlet Problem for the Fractional Laplacian

We begin by rewriting the system (4.1.3b) in a more general form. That is,

$$\begin{cases} (-\Delta)^s u = f & \text{in } \Omega, \\ u = z & \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.3.1}$$

Here is our notion of weak solution.

Definition 4.3.1 (Weak solution to the Dirichlet problem). Let $f \in W^{-s,2}(\overline{\Omega})$, $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$ and $\widetilde{z} \in W^{s,2}(\mathbb{R}^N)$ be such that $\widetilde{z}|_{\mathbb{R}^N \setminus \Omega} = z$. A function $u \in W^{s,2}(\mathbb{R}^N)$ is said to be a weak solution to (4.3.1) if $u - \widetilde{z} \in W_0^{s,2}(\overline{\Omega})$ and

$$\frac{C_{N,s}}{2} \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{(u(x) - u(y))(v(x) - v(y))}{|x - y|^{N+2s}} \, dxdy = \langle f, v \rangle,$$

for every $v \in W_0^{s,2}(\overline{\Omega})$.

Firstly, we notice that since Ω is assumed to have a Lipschitz continuous boundary, we have that, for $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$, there exists $\widetilde{z} \in W^{s,2}(\mathbb{R}^N)$ such that $\widetilde{z}|_{\mathbb{R}^N \setminus \Omega} = z$. Secondly, the existence and uniqueness of a weak solution u to (4.3.1) and the continuous dependence of u on the data f and z have been considered in [73] (see also [65,143]). More precisely we have the following result.

Proposition 4.3.2. Let $f \in W^{-s,2}(\overline{\Omega})$ and $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$. Then there exists a unique weak solution u to (4.3.1) in the sense of Theorem 4.3.1. In addition there is a constant $C > 0$ such that

$$\|u\|_{W^{s,2}(\mathbb{R}^N)} \leq C \left(\|f\|_{W^{-s,2}(\overline{\Omega})} + \|z\|_{W^{s,2}(\mathbb{R}^N \setminus \Omega)} \right). \tag{4.3.2}$$

Even though such a result is typically sufficient in most situations, nevertheless it is not directly useful in the current context of optimal control problem (4.1.3) since we are interested in taking the space $Z_D = L^2(\mathbb{R}^N \setminus \Omega)$. Thus we need existence of solutions (in some sense) to the fractional Dirichlet problem (4.3.1) when $z \in L^2(\mathbb{R}^N \setminus \Omega)$. In order to tackle this situation we introduce our notion of very-weak solution for (4.3.1).

Definition 4.3.3 (Very-Weak Solution to the Dirichlet Problem). Let $z \in L^2(\mathbb{R}^N \setminus \Omega)$ and $f \in W^{-s,2}(\overline{\Omega})$. A function $u \in L^2(\mathbb{R}^N)$ is said to be a very-weak solution to (4.3.1) if the identity

$$\int_\Omega u(-\Delta)^s v \, dx = \langle f, v \rangle - \int_{\mathbb{R}^N \setminus \Omega} z \mathcal{N}_s v \, dx, \tag{4.3.3}$$

holds for every $v \in V := \{v \in W_0^{s,2}(\overline{\Omega}) \; : \; (-\Delta)^s v \in L^2(\Omega)\}$.

Remark 4.3.4. *We mention the following facts.*

(i) We have shown in Theorem 4.3.2 that if $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$, then the Dirichlet problem (4.3.1) has a unique weak solution $u \in W^{s,2}(\mathbb{R}^N)$. In [63], letting

$$\mathcal{V}(\Omega) := \left\{ v : \mathbb{R}^N \to \mathbb{R}, \; v \in L^2(\Omega) \;\; \text{and} \;\; \int_\Omega \int_{\mathbb{R}^N} \frac{|v(x) - v(y)|^2}{|x - y|^{N+2s}} \, dxdy < \infty \right\},$$

the authors have shown that if $z \in \mathcal{V}(\Omega)$ and $f \in \mathcal{V}(\Omega)^*$, then the Dirichlet problem (4.3.1) has a unique weak solution $u \in \mathcal{V}(\Omega)$. Notice that $W^{s,2}(\mathbb{R}^N) \hookrightarrow \mathcal{V}(\Omega) \hookrightarrow W^{s,2}(\Omega)$. The difference between the two notions is only the space where the exterior *data and the solutions belong.*

(ii) For the very-weak solution, we have just assumed that $z \in L^2(\mathbb{R}^N \setminus \Omega)$ (no additional regularity) and this has not been studied in [63] or elsewhere.

Next we prove the existence and uniqueness of a very-weak solution to (4.3.1).

Theorem 4.3.5. Let $f \in W^{-s,2}(\overline{\Omega})$ and $z \in L^2(\mathbb{R}^N \setminus \Omega)$. Then there exists a unique very-weak solution u to (4.3.1) according to Theorem 4.3.1 that fulfills

$$\|u\|_{L^2(\Omega)} \le C\left(\|f\|_{W^{-s,2}(\overline{\Omega})} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\right), \tag{4.3.4}$$

for a constant $C > 0$. In addition, if $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$, then the following assertions hold.

(a) Every weak solution of (4.3.1) is also a very-weak solution.

(b) Every very-weak solution of (4.3.1) that belongs to $W^{s,2}(\mathbb{R}^N)$ is also a weak solution.

Proof. In order to show the existence of a very-weak solution we shall apply the Babuška-Lax-Milgram theorem.

Firstly, let $(-\Delta)^s_D$ be the realization of $(-\Delta)^s$ in $L^2(\Omega)$ with the zero Dirichlet exterior condition $u = 0$ in $\mathbb{R}^N \setminus \Omega$. More precisely,

$$D((-\Delta)^s_D) = V \text{ and } (-\Delta)^s_D u = (-\Delta)^s u \text{ in } \Omega.$$

Then a norm on V is given by $\|v\|_V = \|(-\Delta)^s_D v\|_{L^2(\Omega)}$ which follows from the fact that the operator $(-\Delta)^s_D$ is invertible (since by [128] $(-\Delta)^s_D$ has a compact resolvent and its first eigenvalue is strictly positive). Secondly, let $\mathcal{F}$ be the bilinear form defined on $L^2(\Omega) \times V$ by

$$\mathcal{F}(u, v) := \int_\Omega u(-\Delta)^s v \, dx.$$

Then $\mathcal{F}$ is clearly bounded on $L^2(\Omega) \times V$. More precisely there is a constant $C > 0$ such that

$$\big|\mathcal{F}(u,v)\big| \le \|u\|_{L^2(\Omega)}\|(-\Delta)^s v\|_{L^2(\Omega} \le C\|u\|_{L^2(\Omega)}\|v\|_V, \;\; \forall\,(u,v)\in L^2(\Omega)\times V.$$

Thirdly, we show the inf-sup conditions. From the definition of V, we have that

$$v \in W_0^{s,2}(\overline{\Omega}) \;\text{ and }\; (-\Delta)^s v \in L^2(\Omega) \quad\Longleftrightarrow\quad v\in V.$$

Letting $u := \frac{(-\Delta)_D^s v}{\|(-\Delta)_D^s v\|_{L^2(\Omega)}} \in L^2(\Omega)$, we obtain that

$$\sup_{u\in L^2(\Omega),\|u\|_{L^2(\Omega)}=1} |(u,(-\Delta)_D^s v)_{L^2(\Omega)}| \ge \frac{|((-\Delta)_D^s v,(-\Delta)_D^s v)_{L^2(\Omega)}|}{\|(-\Delta)_D^s v\|_{L^2(\Omega)}}$$

$$\ge \|(-\Delta)_D^s v\|_{L^2(\Omega)} = \|v\|_V.$$

Next we choose $v\in V$ as the unique weak solution of the Dirichlet problem

$$(-\Delta)_D^s v = \frac{u}{\|u\|_{L^2(\Omega)}} \quad\text{in }\; \Omega,$$

for some $0\ne u\in L^2(\Omega)$. Then we readily obtain that

$$\sup_{v\in V,\|v\|_V=1} |(u,(-\Delta)^s v)_{L^2(\Omega)}| \ge \frac{|(u,u)_{L^2(\Omega)}|}{\|u\|_{L^2(\Omega)}} = \|u\|_{L^2(\Omega)} > 0,$$

for all $0\ne u\in L^2(\Omega)$. Finally, we have to show that the right-hand-side in (4.3.3) defines a linear continuous functional on V. Indeed, applying the Hölder inequality in conjunction with Theorem 4.2.1 we obtain that there is a constant $C>0$ such that

$$\left|\int_{\mathbb{R}^N\setminus\Omega} z\mathcal{N}_s v\,dx\right| \le \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|\mathcal{N}_s v\|_{L^2(\mathbb{R}^N\setminus\Omega)} \le C\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|v\|_{W_0^{s,2}(\overline{\Omega})}, \tag{4.3.5}$$

where in the last step we have used the fact that $\|v\|_{W_0^{s,2}(\overline{\Omega})} = \|v\|_{W^{s,2}(\mathbb{R}^N)}$ for $v \in W_0^{s,2}(\overline{\Omega})$. Moreover

$$|\langle f, v\rangle| \le \|f\|_{W^{-s,2}(\overline{\Omega})}\|v\|_{W_0^{s,2}(\overline{\Omega})}.$$

In view of the last two estimates, the right-hand-side in (4.3.3) defines a linear continuous functional on V. Therefore all the requirements of the Babuška-Lax-Milgram theorem hold. Thus, there exists a unique $u \in L^2(\Omega)$ satisfying (4.3.3). Let $u = z$ in $\mathbb{R}^N \setminus \Omega$. Then $u \in L^2(\mathbb{R}^N)$ and satisfies (4.3.3). We have shown the existence and uniqueness of a very-weak solution.

Next we show the estimate (4.3.4). Let $u \in L^2(\mathbb{R}^N)$ be a very-weak solution. Let $v \in V$ be a weak solution of $(-\Delta)_D^s v = u$ in Ω. Taking this v as a test function in (4.3.3) and using (4.3.5), we get that there is a constant $C > 0$ such that

$$\begin{aligned}
\|u\|_{L^2(\Omega)}^2 \le& \|f\|_{W^{-s,2}(\overline{\Omega})}\|v\|_{W_0^{s,2}(\overline{\Omega})} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|\mathcal{N}_s v\|_{L^2(\mathbb{R}^N\setminus\Omega)}\\
\le& C\left(\|f\|_{W^{-s,2}(\overline{\Omega})} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\right)\|v\|_{W_0^{s,2}(\overline{\Omega})}\\
\le& C\left(\|f\|_{W^{-s,2}(\overline{\Omega})} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\right)\|(-\Delta)_D^s v\|_{L^2(\Omega)}\\
\le& C\left(\|f\|_{W^{-s,2}(\overline{\Omega})} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\right)\|u\|_{L^2(\Omega)}.
\end{aligned}$$

We have shown (4.3.4) and this completes the proof of the first part.

Next we prove the last two assertions of the theorem. Assume that $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$.

(a) Let $u \in W^{s,2}(\mathbb{R}^N) \hookrightarrow L^2(\mathbb{R}^N)$ be a weak solution of (4.3.1). It follows from the definition that $u = z$ in $\mathbb{R}^N \setminus \Omega$ and

$$\frac{C_{N,s}}{2}\int_{\mathbb{R}^N}\int_{\mathbb{R}^N}\frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy = \langle f, v\rangle, \tag{4.3.6}$$

for every $v \in V$. Since $v = 0$ in $\mathbb{R}^N \setminus \Omega$, we have that

$$\int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy$$

$$= \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy. \tag{4.3.7}$$

Using (4.3.6), (4.3.7), the integration by parts formula (4.2.5) together with the fact that $u = z$ in $\mathbb{R}^N \setminus \Omega$, we get that

$$\frac{C_{N,s}}{2}\int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy$$

$$= \langle f, v\rangle$$

$$= \int_\Omega u(-\Delta)^s v\,dx + \int_{\mathbb{R}^N\setminus\Omega} u\mathcal{N}_s v\,dx$$

$$= \int_\Omega u(-\Delta)^s v\,dx + \int_{\mathbb{R}^N\setminus\Omega} z\mathcal{N}_s v\,dx.$$

Thus u is a very-weak solution of (4.3.1).

(b) Finally let u be a very-weak solution of (4.3.1) and assume that $u \in W^{s,2}(\mathbb{R}^N)$. Since $u = z$ in $\mathbb{R}^N \setminus \Omega$, we have that $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$ and if $\widetilde{z} \in W^{s,2}(\mathbb{R}^N)$ satisfies $\widetilde{z}|_{\mathbb{R}^N\setminus\Omega} = z$, then clearly $(u - \widetilde{z}) \in W_0^{s,2}(\overline{\Omega})$. Since u is a very-weak solution of (4.3.1), then by definition, for every $v \in V = D((-\Delta)^s_D)$, we have

$$\int_\Omega u(-\Delta)^s v\,dx = \langle f, v\rangle - \int_{\mathbb{R}^N\setminus\Omega} z\mathcal{N}_s v\,dx. \tag{4.3.8}$$

Since $u \in W^{s,2}(\mathbb{R}^N)$ and $v = 0$ in $\mathbb{R}^N \setminus \Omega$, then using (4.2.5) again we get that

$$\begin{aligned}
&\int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy \\
&\qquad = \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy \\
&\qquad = \int_\Omega u(-\Delta)^s v\,dx + \int_{\mathbb{R}^N\setminus\Omega} u\mathcal{N}_s v\,dx \\
&\qquad = \int_\Omega u(-\Delta)^s v\,dx + \int_{\mathbb{R}^N\setminus\Omega} z\mathcal{N}_s v\,dx. \qquad (4.3.9)
\end{aligned}$$

It follows from (4.3.8) and (4.3.9) that for every $v \in V$, we have

$$\int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy = \langle f, v\rangle. \qquad (4.3.10)$$

Since V is dense in $W_0^{s,2}(\overline{\Omega})$, we have that (4.3.10) remains true for every $v \in W_0^{s,2}(\overline{\Omega})$. We have shown that u is a weak solution of (4.3.1) and the proof is finished. □

4.3.2 The Robin Problem for the Fractional Laplacian

In order to study the Robin problem (4.1.4b) we consider the Sobolev space introduced in [57]. For $g \in L^1(\mathbb{R}^N \setminus \Omega)$ fixed, we let

$$W^{s,2}_{\Omega,g} := \Big\{u : \mathbb{R}^N \to \mathbb{R} \ \text{ measurable, } \|u\|_{W^{s,2}_{\Omega,g}} < \infty\Big\},$$

where

$$\|u\|_{W^{s,2}_{\Omega,g}} := \left(\|u\|^2_{L^2(\Omega)} + \| \, |g|^{\frac{1}{2}} u\|^2_{L^2(\mathbb{R}^N\setminus\Omega)} + \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy \right)^{\frac{1}{2}}. \tag{4.3.11}$$

Let μ be the measure on $\mathbb{R}^N \setminus \Omega$ given by $d\mu = |g|dx$. With this setting, the norm in (4.3.11) can be rewritten as

$$\|u\|_{W^{s,2}_{\Omega,g}} := \left(\|u\|^2_{L^2(\Omega)} + \|u\|^2_{L^2(\mathbb{R}^N\setminus\Omega,\mu)} + \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy \right)^{\frac{1}{2}}. \tag{4.3.12}$$

If $g = 0$, we shall let $W^{s,2}_{\Omega,0} = W^{s,2}_{\Omega}$. The following result has been proved in [57, Proposition 3.1].

Proposition 4.3.6. Let $g \in L^1(\mathbb{R}^N \setminus \Omega)$. Then $W^{s,2}_{\Omega,g}$ is a Hilbert space.

Throughout the remainder of the article, for $g \in L^1(\mathbb{R}^N \setminus \Omega)$, we shall denote by $(W^{s,2}_{\Omega,g})^\star$ the dual of $W^{s,2}_{\Omega,g}$.

Remark 4.3.7. *We mention the following facts.*

(a) Recall that

$$\mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2 = (\Omega \times \Omega) \cup (\Omega \times (\mathbb{R}^N \setminus \Omega)) \cup ((\mathbb{R}^N \setminus \Omega) \times \Omega),$$

so that

$$\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy = \int_\Omega\int_\Omega \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy$$
$$+ \int_\Omega\int_{\mathbb{R}^N\setminus\Omega} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy + \int_{\mathbb{R}^N\setminus\Omega}\int_\Omega \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} dxdy. \tag{4.3.13}$$

(b) If $g \in L^1(\mathbb{R}^N \setminus \Omega)$ and $u \in W^{s,2}_{\Omega,g}$, then using the Hölder inequality we get that

$$\left|\int_{\mathbb{R}^N\setminus\Omega} gu\,dx\right| \leq \int_{\mathbb{R}^N\setminus\Omega} |g|^{\frac{1}{2}}|g^{\frac{1}{2}}|u|\,dx \leq \left(\int_{\mathbb{R}^N\setminus\Omega}|g|\,dx\right)^{\frac{1}{2}}\left(\int_{\mathbb{R}^N\setminus\Omega}|gu^2|\,dx\right)^{\frac{1}{2}}$$

$$\leq \|g\|^{\frac{1}{2}}_{L^1(\mathbb{R}^N\setminus\Omega)}\|u\|_{L^2(\mathbb{R}^N\setminus\Omega,\mu)} \leq \|g\|^{\frac{1}{2}}_{L^1(\mathbb{R}^N\setminus\Omega)}\|u\|_{W^{s,2}_{\Omega,g}}. \tag{4.3.14}$$

It follows from (4.3.14) that in particular, $L^1(\mathbb{R}^N \setminus \Omega, \mu) \hookrightarrow (W^{s,2}_{\Omega,g})^\star$.

(c) By definition (using also (4.3.13)), $W^{s,2}_{\Omega,g} \hookrightarrow W^{s,2}_{\Omega} \hookrightarrow W^{s,2}(\Omega)$, so that we have the following continuous embeddings:

$$W^{s,2}_{\Omega,g} \hookrightarrow W^{s,2}_{\Omega} \hookrightarrow L^{\frac{2N}{N-2s}}(\Omega). \tag{4.3.15}$$

It follows from (4.3.15) that the embeddings $W^{s,2}_{\Omega,g} \hookrightarrow L^2(\Omega)$ and $W^{s,2}_{\Omega} \hookrightarrow L^2(\Omega)$ are compact.

We consider a generalized version of the system (4.1.4b) with nonzero right-hand-side f. That is, the problem:

$$\begin{cases} (-\Delta)^s u = f & \text{in } \Omega, \\ \mathcal{N}_s u + \kappa u = \kappa z & \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.3.16}$$

Throughout the following sections, the measure μ is defined with g replaced by κ. That is, $d\mu = \kappa dx$ (recall that κ is assumed to be non-negative). Here is our notion of weak solution.

Definition 4.3.8. Let $z \in L^2(\mathbb{R}^N \setminus \Omega, \mu)$ and $f \in (W^{s,2}_{\Omega,\kappa})^\star$. A function $u \in W^{s,2}_{\Omega,\kappa}$ is said to be a weak solution of (4.3.16) if the identity

$$\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy + \int_{\mathbb{R}^N\setminus\Omega} \kappa uv\,dx$$

$$= \langle f, v\rangle_{(W^{s,2}_{\Omega,\kappa})^\star, W^{s,2}_{\Omega,\kappa}} + \int_{\mathbb{R}^N\setminus\Omega} \kappa zv\,dx, \tag{4.3.17}$$

holds for every $v \in W^{s,2}_{\Omega,\kappa}$.

We have the following existence result.

Proposition 4.3.9. Let $\kappa \in L^1(\mathbb{R}^N \setminus \Omega) \cap L^\infty(\mathbb{R}^N \setminus \Omega)$. Then for every $z \in L^2(\mathbb{R}^N \setminus \Omega, \mu)$ and $f \in (W^{s,2}_{\Omega,\kappa})^\star$, there exists a weak solution $u \in W^{s,2}_{\Omega,\kappa}$ of (4.3.16).

Proof. Let $D(\mathcal{E}) = W^{s,2}_{\Omega,\kappa}$ and $\mathcal{E} : D(\mathcal{E}) \times D(\mathcal{E}) \to \mathbb{R}$ be given by

$$\mathcal{E}(u,v) := \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy + \int_{\mathbb{R}^N\setminus\Omega} \kappa uv\,dx. \tag{4.3.18}$$

Then $\mathcal{E}$ is a bilinear, symmetric, continuous and closed form on $L^2(\Omega)$. Hence, for every $z \in L^2(\mathbb{R}^N \setminus \Omega, \mu) \subset (W^{s,2}_{\Omega,\kappa})^\star$ and $f \in (W^{s,2}_{\Omega,\kappa})^\star$, there is a function $u \in W^{s,2}_{\Omega,\kappa}$ such that

$$\begin{aligned}\mathcal{E}(u,v) =& \langle f, v\rangle_{(W^{s,2}_{\Omega,\kappa})^\star, W^{s,2}_{\Omega,\kappa}} + \langle z, v\rangle_{(W^{s,2}_{\Omega,\kappa})^\star, W^{s,2}_{\Omega,\kappa}} \\ =& \langle f, v\rangle_{(W^{s,2}_{\Omega,\kappa})^\star, W^{s,2}_{\Omega,\kappa}} + \int_{\mathbb{R}^N\setminus\Omega} \kappa zv\,dx,\end{aligned}$$

for every $v \in W^{s,2}_{\Omega,\kappa}$. That is, u satisfies (4.3.17). Thus u is a weak solution of (4.3.16). The proof is finished. □

Remark 4.3.10. Notice that similarly to the classical Neumann problem when $\kappa \equiv 0$, Theorem 4.3.9 only guarantees uniqueness of solutions to (4.1.4b) up to a constant. In case we assume κ to be strictly positive, uniqueness can be guaranteed under Theorem 4.6.1

below. In that case we can also show that there is a constant $C > 0$ such that

$$\|u\|_{W^{s,2}_{\Omega,\kappa}} \leq C\left(\|f\|_{(W^{s,2}_{\Omega,\kappa})^\star} + \|z\|_{L^2(\mathbb{R}^N\setminus\Omega,\mu)}\right). \tag{4.3.19}$$

4.4 Fractional Dirichlet Exterior Control Problem

We begin by introducing the appropriate function spaces needed to study (4.1.3). We let

$$Z_D := L^2(\mathbb{R}^N \setminus \Omega), \quad U_D := L^2(\Omega).$$

In view of Theorem 4.3.5 the following (solution-map) control-to-state map

$$S : Z_D \to U_D, \;\; z \mapsto Sz = u,$$

is well-defined, linear and continuous. We also notice that for $z \in Z_D$, we have that $u := Sz \in L^2(\mathbb{R}^N)$. As a result we can write the *reduced fractional Dirichlet exterior control problem* as follows:

$$\min_{z\in Z_{ad,D}} \mathcal{J}(z) := J(Sz) + \frac{\xi}{2}\|z\|^2_{Z_D}, \tag{4.4.1}$$

where $\xi \geq 0$. The precise conditions on J depend on the result we would like to obtain. For this reason they will be given in the statements of our results.

We then have the following well-posedness result for (4.4.1) and equivalently (4.1.3).

Theorem 4.4.1. Let $Z_{ad,D}$ be a closed and convex subset of Z_D. Let either $\xi > 0$ or $Z_{ad,D}$ bounded and let $J : U_D \to \mathbb{R}$ be weakly lower-semicontinuous. Then there exists a solution $\bar{z}$ to (4.4.1) and equivalently to (4.1.3). If either J is convex and $\xi > 0$ or J is strictly convex and $\xi \geq 0$, then $\bar{z}$ is unique.

Proof. The proof uses the so-called direct-method or the Weierstrass theorem [25, Theorem 3.2.1]. We notice that for $\mathcal{J} : Z_{ad,D} \to \mathbb{R}$, we can construct a minimizing sequence

$\{z_n\}_{n\in\mathbb{N}}$ (cf. [25, Theorem 3.2.1] for a construction) such that

$$\inf_{z\in Z_{ad,D}} \mathcal{J}(z) = \lim_{n\to\infty} \mathcal{J}(z_n).$$

If $\xi > 0$ or $Z_{ad,D} \subset Z_D$ is bounded, then $\{z_n\}_{n\in\mathbb{N}}$ is a bounded sequence in Z_D which is a Hilbert space. Due to the reflexivity of Z_D, we have that (up to a subsequence if necessary) $z_n \rightharpoonup \bar{z}$ (weak convergence) in Z_D as $n \to \infty$. Since $Z_{ad,D}$ is closed and convex, hence is weakly closed, we have that $\bar{z} \in Z_{ad,D}$.

Since $S : Z_{ad,D} \to U_D$ is linear and continuous, we have that it is weakly continuous. This implies that $Sz_n \rightharpoonup S\bar{z}$ in U_D as $n \to \infty$. We have to show that $(S\bar{z}, \bar{z})$ fulfills the state equation according to Theorem 4.3.3. In particular we need to study the identity

$$\int_\Omega u_n(-\Delta)^s v\, dx = -\int_{\mathbb{R}^N\setminus\Omega} z_n \mathcal{N}_s v\, dx, \quad \forall\, v \in V, \tag{4.4.2}$$

as $n \to \infty$, where $u_n := Sz_n$. Since $u_n \rightharpoonup S\bar{z} =: \bar{u}$ in U_D as $n \to \infty$ and $z_n \rightharpoonup \bar{z}$ in Z_D as $n \to \infty$, we can immediately take the limit in (4.4.2) to obtain that

$$\int_\Omega \bar{u}(-\Delta)^s v\, dx = -\int_{\mathbb{R}^N\setminus\Omega} \bar{z} \mathcal{N}_s v\, dx, \quad \forall\, v \in V.$$

Thus $(\bar{u}, \bar{z}) \in U_D \times Z_{ad,D}$ fulfills the state equation in the sense of Theorem 4.3.3.

It then remains to show that $\bar{z}$ is the minimizer of (4.4.1). This is a consequence of the fact that $\mathcal{J}$ is weakly lower semicontinuous. Indeed, $\mathcal{J}$ is the sum of two weakly lower semicontinuous functions ($\|\cdot\|^2_{Z_D}$ is continuous and convex therefore weakly lower semicontinuous).

Finally, if $\xi > 0$ and J is convex, then $\mathcal{J}$ is strictly convex (sum of a strictly convex and convex functions). On the other hand, if J is strictly convex, then $\mathcal{J}$ is strictly convex. In either case we have that $\mathcal{J}$ is strictly convex and thus the uniqueness of $\bar{z}$ follows. □

We next derive the first order necessary optimality conditions for (4.4.1). We begin by identifying the structure of the adjoint operator S^*.

Lemma 4.4.2. For the state equation (4.1.3b) the adjoint operator $S^* : U_D \to Z_D$ is given by

$$S^* w = -\mathcal{N}_s p \in Z_D,$$

where $w \in U_D$ and $p \in W_0^{s,2}(\overline{\Omega})$ is the weak solution to the problem

$$\begin{cases} (-\Delta)^s p & = w \quad \text{ in } \Omega, \\ p & = 0 \quad \text{ in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.4.3}$$

Proof. According to the definition of S^*, we have that for every $w \in U_D$ and $z \in Z_D$,

$$(w, Sz)_{L^2(\Omega)} = (S^* w, z)_{L^2(\mathbb{R}^N \setminus \Omega)}.$$

Next, testing the adjoint equation (4.4.3) with Sz and using the fact that Sz is a very-weak solution of (4.3.1) with $f = 0$, we arrive at

$$(w, Sz)_{L^2(\Omega)} = (Sz, (-\Delta)^s p)_{L^2(\Omega)} = -(z, \mathcal{N}_s p)_{L^2(\mathbb{R}^N \setminus \Omega)} = (z, S^* w)_{L^2(\mathbb{R}^N \setminus \Omega)}.$$

This yields the asserted result. □

For the remainder of this section we will assume that $\xi > 0$.

Theorem 4.4.3. Let the assumptions of Theorem 4.4.1 hold. Let $\mathcal{Z}$ be an open set in Z_D such that $Z_{ad,D} \subset \mathcal{Z}$. Let $u \mapsto J(u) : U_D \to \mathbb{R}$ be continuously Fréchet differentiable with $J'(u) \in U_D$. If $\bar{z}$ is a minimizer of (4.4.1) over $Z_{ad,D}$, then the first order necessary optimality conditions are given by

$$(-\mathcal{N}_s \bar{p} + \xi \bar{z}, z - \bar{z})_{L^2(\mathbb{R}^N \setminus \Omega)} \geq 0, \quad \forall z \in Z_{ad,D}, \tag{4.4.4}$$

where $\bar{p} \in W_0^{s,2}(\overline{\Omega})$ solves the adjoint equation

$$\begin{cases} (-\Delta)^s \bar{p} = J'(\bar{u}) & \text{in } \Omega, \\ \bar{p} = 0 & \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.4.5}$$

Equivalently we can write (4.4.4) as

$$\bar{z} = \mathcal{P}_{Z_{ad,D}} \left(\frac{1}{\xi} \mathcal{N}_s \bar{p} \right), \tag{4.4.6}$$

where $\mathcal{P}_{Z_{ad,D}}$ is the projection onto the set $Z_{ad,D}$. If J is convex, then (4.4.4) is a sufficient condition.

Proof. The proof is a straightforward application of the differentiability properties of J and the chain rule in conjunction with Theorem 4.4.2. Indeed, for a given direction $h \in Z_{ad,D}$, the directional derivative of $\mathcal{J}$ is given by

$$\begin{aligned} \mathcal{J}'(\bar{z})h =& (J'(S\bar{z}), Sh)_{L^2(\Omega)} + \xi(\bar{z}, h)_{L^2(\mathbb{R}^N \setminus \Omega)} \\ =& (S^* J'(S\bar{z}), h)_{L^2(\Omega)} + \xi(\bar{z}, h)_{L^2(\mathbb{R}^N \setminus \Omega)}, \end{aligned}$$

where in the first step we have used that $J'(S\bar{z}) \in \mathcal{L}(L^2(\Omega), \mathbb{R}) = L^2(\Omega)$ and in the second step we have used that S is linear and bounded, therefore S^* is well-defined. Then using Theorem 4.4.2 we arrive at the asserted result. From Theorem 4.2.1 we recall that $\mathcal{N}_s \bar{p} \in L^2(\mathbb{R}^N \setminus \Omega)$. Therefore the equivalence between (4.4.4) and (4.4.6) follows by using [25, Theorem 3.3.5]. The proof is finished. □

Remark 4.4.4 (**Regularity for the Optimization Variables).** We recall a rather surprising result for the adjoint equation (4.4.3). The standard maximal elliptic regularity that is known to hold for the classical Laplacian on smooth open sets does not hold in the case of the fractional Laplacian i.e., p does not always belong to $W^{2s,2}(\Omega)$. Notice that $w \in L^2(\Omega)$ and $p = [(-\Delta)^s_D]^{-1}w$. More precisely assume that Ω is a smooth bounded open set. If $0 < s < \frac{1}{2}$, then by [73, Formula (7.4)] we have that $D((-\Delta)^s_D) = W_0^{2s,2}(\overline{\Omega})$ and hence, $p \in W^{2s,2}(\Omega)$ in that case. But if $\frac{1}{2} \le s < 1$, an example has been given in [120, Remark 7.2] where $D((-\Delta)^s_D) \not\subset W^{2s,2}(\Omega)$, thus in that case p does not always belong to $W^{2s,2}(\Omega)$. It has been shown in [33, 34] that only a local maximal elliptic regularity can be achieved. As a result, the best known result for $\mathcal{N}_s p$ is as given in Theorem 4.2.1. Since $\mathcal{P}_{Z_{ad,D}}$ is a contraction (Lipschitz) we can conclude that $\bar{z}$ has the same regularity as $\mathcal{N}_s\bar{p}$, i.e., they are in $L^2(\mathbb{R}^N \setminus \Omega)$ globally and in $W^{s,2}_{\text{loc}}(\mathbb{R}^N \setminus \Omega)$ locally. As it is well-known, in case of the classical Laplacian, one can use a boot-strap argument to improve the regularity of $S\bar{z} = \bar{u}$ globally. However this is not the case for the fractional exterior value problems. We also notice that always for the case $\frac{1}{2} < s < 1$, Grubb [72, Section 2] (see also [73]) has introduced some fractional order Sobolev spaces where a maximal elliptic regularity is obtained on these spaces. Of course these fractional order Sobolev spaces do not coincide with $W^{2s,2}(\Omega)$. She has also proved some maximal elliptic regularity on some certain spaces of Hölder continuous functions. We recall that our operator $(-\Delta)^s_D$ is different from the spectral Dirichlet fractional Laplacian (the fractional powers of the Laplace operator with the Dirichlet boundary condition). For the latter operator, Grubb [74] has shown that a maximal elliptic regularity can be achieved in some classical fractional order Sobolev spaces.

4.5 Fractional Robin Exterior Control Problem

In this section we study the fractional Robin exterior control problem (4.1.4b). We begin by setting the functional analytic framework. We let

$$Z_R := L^2(\mathbb{R}^N \setminus \Omega, \mu), \quad U_R := W^{s,2}_{\Omega,\kappa}.$$

Notice that $d\mu = \kappa dx$. In addition we assume that $\kappa \in L^1(\mathbb{R}^N \setminus \Omega) \cap L^\infty(\mathbb{R}^N \setminus \Omega)$ and $\kappa > 0$ a.e. in $\mathbb{R}^N \setminus \Omega$. In view of Theorem 4.3.9 the following (solution-map) control-to-state map

$$S : Z_R \to U_R, \quad z \mapsto u,$$

is well-defined. Moreover S is linear and continuous (by (4.3.19)). Since $U_R \hookrightarrow L^2(\Omega)$ with the embedding being continuous we can instead define

$$S : Z_R \to L^2(\Omega).$$

We can then write the so-called *reduced fractional Robin exterior control problem* as follows:

$$\min_{z \in Z_{ad,R}} \mathcal{J}(z) := J(Sz) + \frac{\xi}{2} \|z\|^2_{L^2(\mathbb{R}^N \setminus \Omega, \mu)}, \tag{4.5.1}$$

where $\xi \geq 0$. Here also, the precise conditions on J will be given in the statements of the results. We have the following well-posedness result.

Theorem 4.5.1. Let $Z_{ad,R}$ be a closed and convex subset of Z_R. Let either $\xi > 0$ or $Z_{ad,R} \subset Z_R$ bounded. Moreover, let $J : L^2(\Omega) \to \mathbb{R}$ be weakly lower-semicontinuous. Then there exists a solution $\bar{z}$ to (4.5.1) and equivalently to (4.1.4). If either J is convex and $\xi > 0$ or J is strictly convex and $\xi \geq 0$ then $\bar{z}$ is unique.

Proof. We proceed as in the proof of Theorem 4.4.1. Let $\{z_n\}_{n\in\mathbb{N}} \subset Z_{ad,R}$ be a minimizing sequence such that

$$\inf_{z \in Z_{ad,R}} \mathcal{J}(z) = \lim_{n\to\infty} \mathcal{J}(z_n).$$

If $\xi > 0$ or $Z_{ad,R} \subset Z_R$ is bounded, then after a subsequence, if necessary, we have that $z_n \rightharpoonup \bar{z}$ in $L^2(\mathbb{R}^N \setminus \Omega, \mu)$ as $n \to \infty$. Now since $Z_{ad,R}$ is a convex and closed subset of Z_R, it follows that $\bar{z} \in Z_{ad,R}$.

Next we show that the pair $(S\bar{z}, \bar{z})$ satisfies the state equation. Notice that $u_n := Sz_n$ is the weak solution of (4.1.4b) with boundary value z_n. Thus, by definition, $u_n \in W^{s,2}_{\Omega,\kappa}$ and the identity

$$\mathcal{E}(u_n, v) = \int_{\mathbb{R}^N \setminus \Omega} z_n v \, d\mu, \tag{4.5.2}$$

holds for every $v \in W^{s,2}_{\Omega,\kappa}$ where we recall that $\mathcal{E}$ is given in (4.3.18). We also notice that the mapping S is also bounded from Z_R into $W^{s,2}_{\Omega,\kappa}$ (by (4.3.19)). This shows that the sequence $\{u_n\}_{n\in\mathbb{N}}$ is bounded in $W^{s,2}_{\Omega,\kappa}$. Thus, after a subsequence, if necessary, we have that $Sz_n = u_n \rightharpoonup S\bar{z} = \bar{u}$ in $W^{s,2}_{\Omega,\kappa}$ as $n \to \infty$. This implies that

$$\lim_{n\to\infty} \left(\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u_n(x) - u_n(y))(v(x) - v(y))}{|x-y|^{N+2s}} \, dxdy + \int_{\mathbb{R}^N\setminus\Omega} u_n v \, d\mu \right)$$
$$= \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(\bar{u}(x) - \bar{u}(y))(v(x) - v(y))}{|x-y|^{N+2s}} \, dxdy + \int_{\mathbb{R}^N\setminus\Omega} \bar{u} v \, d\mu,$$

for every $v \in W^{s,2}_{\Omega,\kappa}$. Since $z_n \rightharpoonup \bar{z}$ in $L^2(\mathbb{R}^N \setminus \Omega, \mu)$ as $n \to \infty$, it follows that

$$\lim_{n\to\infty} \int_{\mathbb{R}^N\setminus\Omega} z_n v \, d\mu = \int_{\mathbb{R}^N\setminus\Omega} \bar{z} v \, d\mu,$$

for every $v \in W^{s,2}_{\Omega,\kappa}$. Therefore we can pass to the limit in (4.5.2) as $n \to \infty$ to obtain that

$$\mathcal{E}(\bar{u}, v) = \int_{\mathbb{R}^N\setminus\Omega} \bar{z} v \, dx, \quad \forall \, v \in W^{s,2}_{\Omega,\kappa}.$$

Thus, $(S\bar{z}, \bar{z}) = (\bar{u}, \bar{z})$ satisfies the state equation (4.1.4b). The rest of the steps are similar to the proof of Theorem 4.4.1 and we omit them for brevity. □

As in the case of the fractional Dirichlet exterior control problem (4.4.1) we identify next the adjoint of the control-to-state map S.

Lemma 4.5.2. For the state equation (4.1.4b) the adjoint operator $S^* : L^2(\Omega) \to Z_R$ is given by

$$(S^*w, z)_{Z_R} = \int_{\mathbb{R}^N \setminus \Omega} pz \, d\mu \quad \forall z \in Z_R,$$

where $w \in L^2(\Omega)$ and $p \in W^{s,2}_{\Omega,\kappa}$ is the weak solution to

$$\begin{cases} (-\Delta)^s p &= w \quad \text{in } \Omega, \\ \mathcal{N}_s p + \kappa p &= 0 \quad \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.5.3}$$

Proof. Let $w \in L^2(\Omega)$ and $z \in Z_R$. Then $Sz \in W^{s,2}_{\Omega,\kappa} \hookrightarrow L^2(\Omega)$ with the embedding being continuous. Then we can write

$$(w, Sz)_{L^2(\Omega)} = (S^*w, z)_{Z_R}.$$

Next we test (4.5.3) with Sz to arrive at

$$\begin{aligned} (w, Sz)_{L^2(\Omega)} =& \frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2} \frac{(u(x) - u(y))(p(x) - p(y))}{|x - y|^{N+2s}} \, dxdy + \int_{\mathbb{R}^N \setminus \Omega} up \, d\mu \\ =& \int_{\mathbb{R}^N \setminus \Omega} zp \, d\mu = (S^*w, z)_{Z_R}, \end{aligned}$$

where we have used the fact that u solves the state equation according to Theorem 4.3.8. The proof is finished. □

For the remainder of this section we will assume that $\xi > 0$. The proof of the next result is similar to the proof of Theorem 4.4.3 and is omitted for brevity.

Theorem 4.5.3. Let the assumptions of Theorem 4.5.1 hold. Let $\mathcal{Z}$ be an open set in Z_R such that $Z_{ad,R} \subset \mathcal{Z}$. Let $u \mapsto J(u) : L^2(\Omega) \to \mathbb{R}$ be continuously Fréchet differentiable with $J'(u) \in L^2(\Omega)$. If $\bar{z}$ is a minimizer of (4.5.1) over $Z_{ad,R}$, then the first necessary optimality conditions are given by

$$\int_{\mathbb{R}^N \setminus \Omega} (\bar{p} + \xi \bar{z})(z - \bar{z}) \, d\mu \geq 0, \quad z \in Z_{ad,R}, \tag{4.5.4}$$

where $\bar{p} \in W^{s,2}_{\Omega,\kappa}$ solves the adjoint equation

$$\begin{cases} (-\Delta)^s \bar{p} = J'(\bar{u}) & \text{in } \Omega, \\ \mathcal{N}_s \bar{p} + \kappa \bar{p} = 0 & \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.5.5}$$

Equivalently we can write (4.5.4) as

$$\bar{z} = \mathcal{P}_{Z_{ad,R}} \left(-\frac{\bar{p}}{\xi} \right), \tag{4.5.6}$$

where $\mathcal{P}_{Z_{ad,R}}$ is the projection onto the set $Z_{ad,R}$. If J is convex, then (4.5.4) is a sufficient condition.

Remark 4.5.4 (Regularity of the Optimization Variables). As pointed out in Theorem 4.4.4 (Dirichlet case) the regularity for the integral fractional Laplacian is a delicate issue. In fact for the Robin problem, in $\mathbb{R}^N \setminus \Omega$ we can only guarantee that $\bar{p} \in L^2(\mathbb{R}^N \setminus \Omega, \mu)$. We further emphasize that the regularity for the fractional Robin problem is still open. Therefore due to the lack of such regularity results, we cannot use the classical boot-strap argument to further improve the regularity of the control $\bar{z}$.

4.6 Approximation of Dirichlet Exterior Value and Control Problems

We recall that the Dirichlet exterior value problem (4.1.2) in our case is only understood in the very-weak sense (cf. Theorem 4.3.5). Moreover a numerical approximation of solutions to this problem will require a direct approximation of the interaction operator $\mathcal{N}_s$ which is challenging. Similar situations arise in the first order optimality conditions for the Dirichlet control problem (4.4.4).

The purpose of this section is to not only introduce a new approach to approximate weak and very-weak solutions to the nonhomogeneous Dirichlet exterior value problem (recall that if z is regular enough then a very-weak solution is a weak solution, and every weak solution is a very-weak solution, cf. Theorem 4.3.5) but also to consider a regularized fractional Dirichlet exterior control problem. We begin by stating the regularized Dirichlet exterior value problem. Let $n \in \mathbb{N}$. Find $u_n \in W^{s,2}_{\Omega,\kappa}$ solving the elliptic problem

$$\begin{cases} (-\Delta)^s u_n = 0 & \text{in } \Omega, \\ \mathcal{N}_s u_n + n\kappa u_n = n\kappa z & \text{in } \mathbb{R}^N \setminus \Omega. \end{cases} \tag{4.6.1}$$

Notice that the fractional regularized Dirichlet exterior problem (4.6.1) is nothing but the fractional Robin exterior value problem (4.1.4b). We proceed by showing that the solution u_n to (4.6.1) converges to a function u, as $n \to \infty$, that solves the state equation (4.1.2) in the very weak sense (4.3.3). This is our new method to solve the non-homogeneous Dirichlet exterior value problem. Recall that the weak formulation of (4.6.1) does not require access to $\mathcal{N}_s$ (cf. Definition Theorem 4.3.8) and it is straightforward to implement.

In this section we are interested in solutions u_n to (4.6.1) that belong to $W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$

which is endowed with the norm

$$\|u\|_{W^{s,2}_{\Omega,\kappa}\cap L^2(\mathbb{R}^N\setminus\Omega)} := \left(\|u\|^2_{W^{s,2}_{\Omega,\kappa}} + \|u\|^2_{L^2(\mathbb{R}^N\setminus\Omega)}\right)^{\frac{1}{2}}. \tag{4.6.2}$$

In addition, in our application we shall take κ such that its support $\operatorname{supp}[\kappa] \subset \mathbb{R}^N \setminus \Omega$ has a positive Lebesgue measure. For this reason we shall assume the following.

Assumption 4.6.1. We assume that $\kappa \in L^1(\mathbb{R}^N \setminus \Omega) \cap L^\infty(\mathbb{R}^N \setminus \Omega)$ and satisfies $\kappa > 0$ almost everywhere in $K := \operatorname{supp}[\kappa] \subset \mathbb{R}^N \setminus \Omega$, where the Lebesgue measure $|K| > 0$.

It follows from Theorem 4.6.1 that $\displaystyle\int_{\mathbb{R}^N\setminus\Omega} \kappa \, dx > 0$.

To show the existence of weak solutions to the system (4.6.1) that belong to $W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$, we need some preparation.

Lemma 4.6.2. Assume that Theorem 4.6.1 holds. Then

$$\|u\|_W := \left(\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u(x)-u(y)|^2}{|x-y|^{N+2s}} \, dxdy + \int_{\mathbb{R}^N\setminus\Omega} |u|^2 \, dx\right)^{\frac{1}{2}} \tag{4.6.3}$$

defines an equivalent norm on $W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$.

Proof. Firstly, it is readily seen that there is a constant $C > 0$ such that

$$\|u\|_W \leq C\|u\|_{W^{s,2}_{\Omega,\kappa}\cap L^2(\mathbb{R}^N\setminus\Omega)} \text{ for all } u \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega). \tag{4.6.4}$$

Secondly, we claim that there is a constant $C > 0$ such that

$$\|u\|_{W^{s,2}_{\Omega,\kappa}\cap L^2(\mathbb{R}^N\setminus\Omega)} \leq C\|u\|_W \text{ for all } u \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega). \tag{4.6.5}$$

It is clear that

$$\int_{\mathbb{R}^N\setminus\Omega} |u|^2\, d\mu \leq \|\kappa\|_{L^\infty(\mathbb{R}^N\setminus\Omega)} \int_{\mathbb{R}^N\setminus\Omega} |u|^2\, dx. \tag{4.6.6}$$

It suffices to show that there is a constant $C > 0$ such that for every $u \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$,

$$\int_\Omega |u|^2\, dx \leq C \left(\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u(x) - u(y)|^2}{|x-y|^{N+2s}}\, dxdy + \int_{\mathbb{R}^N\setminus\Omega} |u|^2\, dx \right). \tag{4.6.7}$$

We prove (4.6.7) by contradiction. Assume to the contrary that for every $n \in \mathbb{N}$, there exists a sequence $\{u_n\}_{n\in\mathbb{N}} \subset W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ such that

$$\int_\Omega |u_n|^2\, dx > n \left(\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u_n(x) - u_n(y)|^2}{|x-y|^{N+2s}}\, dxdy + \int_{\mathbb{R}^N\setminus\Omega} |u_n|^2\, dx \right). \tag{4.6.8}$$

By possibly dividing (4.6.8) by $\|u_n\|^2_{L^2(\Omega)}$ we may assume that $\|u_n\|^2_{L^2(\Omega)} = 1$ for every $n \in \mathbb{N}$. Hence, by (4.6.8), there is a constant $C > 0$ (independent of n) such that for every $n \in \mathbb{N}$,

$$\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u_n(x) - u_n(y)|^2}{|x-y|^{N+2s}}\, dxdy + \int_{\mathbb{R}^N\setminus\Omega} |u_n|^2\, dx \leq C. \tag{4.6.9}$$

Since $\kappa \in L^\infty(\mathbb{R}^N \setminus \Omega)$, (4.6.9) and (4.6.6) imply that for every $n \in \mathbb{N}$,

$$\int_{\mathbb{R}^N\setminus\Omega} |u_n|^2\, d\mu \leq C. \tag{4.6.10}$$

Now (4.6.9), (4.6.10) together with $\|u_n\|^2_{L^2(\Omega)} = 1$ imply that $\{u_n\}_{n\in\mathbb{N}}$ is a bounded sequence in the space $W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$. Therefore, after passing to a subsequence, if necessary, we may assume that u_n converges weakly to some $u \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ and strongly to u in

$L^2(\Omega)$, as $n \to \infty$ (as the embedding $W^{s,2}_{\Omega,\kappa} \hookrightarrow L^2(\Omega)$ is compact by Theorem 4.3.7(c)). It follows from (4.6.8) and the fact that $\|u_n\|^2_{L^2(\Omega)} = 1$ that

$$\lim_{n\to\infty} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u_n(x)-u_n(y)|^2}{|x-y|^{N+2s}}\,dxdy = 0 \text{ and } \lim_{n\to\infty}\int_{\mathbb{R}^N\setminus\Omega} |u_n|^2\,dx = 0.$$

These identities imply that $u_n|_{\mathbb{R}^N\setminus\Omega}$ converges strongly to zero in $L^2(\mathbb{R}^N \setminus \Omega)$ as $n \to \infty$, and after passing to a subsequence, if necessary, we have that

$$\lim_{n\to\infty} |u_n(x) - u_n(y)| = 0 \quad \text{for a.e. } (x,y) \in \mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2, \tag{4.6.11}$$

and

$$u_n \to 0 \text{ a.e. in } \mathbb{R}^N \setminus \Omega \text{ as } n \to \infty. \tag{4.6.12}$$

More precisely, (4.6.11) implies that

$$\begin{cases} \lim_{n\to\infty} |u_n(x)-u_n(y)| = 0 & \text{for a.e. } (x,y) \in \Omega\times\Omega, \\ \lim_{n\to\infty} |u_n(x)-u_n(y)| = 0 & \text{for a.e. } (x,y) \in \Omega\times(\mathbb{R}^N\setminus\Omega), \\ \lim_{n\to\infty} |u_n(x)-u_n(y)| = 0 & \text{for a.e. } (x,y) \in (\mathbb{R}^N\setminus\Omega)\times\Omega. \end{cases} \tag{4.6.13}$$

Using (4.6.13), we get that u_n converges a.e. to some constant function c in $\mathbb{R}^N$ as $n \to \infty$. From (4.6.12) and the uniqueness of the limit, we have that $c = 0$ a.e. in $\mathbb{R}^N$. Since (after passing to a subsequence, if necessary) u_n converges a.e. to u in Ω as $n \to \infty$, the uniqueness of the limit also implies that $c = u = 0$ a.e. on Ω. On the other hand, $\|u\|^2_{L^2(\Omega)} = \lim_{n\to\infty} \|u_n\|^2_{L^2(\Omega)} = 1$, and this is a contradiction. Hence, (4.6.8) is not possible and we have shown (4.6.7). Finally the lemma follows from (4.6.4) and (4.6.5). The proof is finished. □

The following theorem is the main result of this section.

Theorem 4.6.3 (**Approximation of Solutions to the Dirichlet Problem).** Assume that Theorem 4.6.1 holds. Then the following assertions hold.

(a) Let $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$ and $u_n \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ be the weak solution of (4.6.1). Let $u \in W^{s,2}(\mathbb{R}^N)$ be the weak solution to the state equation (4.1.3b). Then there is a constant $C > 0$ (independent of n) such that

$$\|u - u_n\|_{L^2(\mathbb{R}^N)} \leq \frac{C}{n}\|u\|_{W^{s,2}(\mathbb{R}^N)}. \tag{4.6.14}$$

In particular u_n converges strongly to u in $L^2(\mathbb{R}^N)$ as $n \to \infty$.

(b) Let $z \in L^2(\mathbb{R}^N \setminus \Omega)$ and $u_n \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ be the weak solution of (4.6.1). Then there exist a subsequence that we still denote by $\{u_n\}_{n\in\mathbb{N}}$ and a function $\tilde{u} \in L^2(\mathbb{R}^N)$ such that $u_n \rightharpoonup \tilde{u}$ in $L^2(\mathbb{R}^N)$ as $n \to \infty$, and $\tilde{u}$ satisfies

$$\int_\Omega \tilde{u}(-\Delta)^s v \, dx = -\int_{\mathbb{R}^N\setminus\Omega} \tilde{u}\mathcal{N}_s v \, dx, \tag{4.6.15}$$

for all $v \in V$.

Remark 4.6.4 (**Convergence to a Very-Weak Solution).** Notice that Part (a) of Theorem 4.6.3 implies strong convergence to a weak solution (with rate). On the other hand, Part (b) "almost" implies weak convergence to a very-weak solution (we still do not know if $\tilde{u}|_{\mathbb{R}^N\setminus\Omega} = z$). We emphasize that such an approximation of very-weak solutions using the Robin problem, to the best of our knowledge, is open even for the classical case $s = 1$ when the boundary function just belongs to $L^2(\partial\Omega)$.

Proof of Theorem 4.6.3. (a) Let $z \in W^{s,2}(\mathbb{R}^N \setminus \Omega)$. Firstly, recall that under our assumption $W^{s,2}(\mathbb{R}^N \setminus \Omega) \hookrightarrow L^2(\mathbb{R}^N \setminus \Omega) \hookrightarrow L^2(\mathbb{R}^N \setminus \Omega, \mu)$. Secondly, consider the system

(4.6.1). A weak solution is a function $u_n \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ such that the identity

$$\frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u_n(x)-u_n(y))(v(x)-v(y))}{|x-y|^{N+2s}}\, dxdy \\ + n\int_{\mathbb{R}^N\setminus\Omega} u_n v\, d\mu = n\int_{\mathbb{R}^N\setminus\Omega} zv\, d\mu, \tag{4.6.16}$$

holds for every $v \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$. Proceeding as in the proof of Theorem 4.3.9 we can easily deduce that for every $n \in \mathbb{N}$, there is a unique $u_n \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ satisfying (4.6.16).

For $v, w \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$ we let

$$\mathcal{E}_n(v,w) := \frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(v(x)-v(y))(w(x)-w(y))}{|x-y|^{N+2s}}\, dxdy + n\int_{\mathbb{R}^N\setminus\Omega} vw\, d\mu.$$

We notice that proceeding as in the proof of Theorem 4.6.2 we can deduce that there is a constant $C > 0$ such that

$$\frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{|u_n(x)-u_n(y)|^2}{|x-y|^{N+2s}}\, dxdy + n\int_{\mathbb{R}^N\setminus\Omega} |u_n|^2\, dx \le C\mathcal{E}_n(u_n,u_n). \tag{4.6.17}$$

Next, let $u \in W^{s,2}(\mathbb{R}^N)$ be the weak solution of (4.3.1) and $v \in W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$.

Using the integration by parts formula (4.2.5) we get that

$$\begin{aligned}\mathcal{E}_n(u-u_n,v) &= \int_\Omega v(-\Delta)^s(u-u_n)\,dx + \int_{\mathbb{R}^N\setminus\Omega} v\mathcal{N}_s(u-u_n)\,dx\\ &\quad + n\int_{\mathbb{R}^N\setminus\Omega} v\,(u-u_n)\;d\mu\\ &= \int_\Omega v(-\Delta)^s(u-u_n)\,dx + \int_{\mathbb{R}^N\setminus\Omega} v\mathcal{N}_s u\,dx\\ &\quad - \int_{\mathbb{R}^N\setminus\Omega} v\left(\mathcal{N}_s u_n + n\kappa(u_n-z)\right)\;dx\\ &= \int_{\mathbb{R}^N\setminus\Omega} v\mathcal{N}_s u\,dx. \end{aligned} \tag{4.6.18}$$

Taking $v = u - u_n$ as a test function in (4.6.18) and using (4.6.17), we get that there is a constant $C > 0$ (independent of n) such that

$$\begin{aligned} n\|u-u_n\|^2_{L^2(\mathbb{R}^N\setminus\Omega)} \le \mathcal{E}_n(u-u_n,u-u_n) &= \int_{\mathbb{R}^N\setminus\Omega}(u-u_n)\mathcal{N}_s u\,dx\\ &\le \|u-u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|\mathcal{N}_s u\|_{L^2(\mathbb{R}^N\setminus\Omega)}\\ &\le C\|u-u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|u\|_{W^{s,2}(\mathbb{R}^N)}. \end{aligned}$$

We have shown that there is a constant $C > 0$ (independent of n) such that

$$\|u-u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)} \le \frac{C}{n}\|u\|_{W^{s,2}(\mathbb{R}^N)}. \tag{4.6.19}$$

Next, observe that

$$\|u-u_n\|_{L^2(\Omega)} = \sup_{\eta\in L^2(\Omega)} \frac{\left|\int_\Omega (u-u_n)\eta\,dx\right|}{\|\eta\|_{L^2(\Omega)}}. \tag{4.6.20}$$

For any $\eta \in L^2(\Omega)$, let $w \in W_0^{s,2}(\overline{\Omega})$ be the weak solution of the Dirichlet problem

$$(-\Delta)^s w = \eta \quad \text{in } \ \Omega, \quad w = 0 \quad \text{in } \ \mathbb{R}^N \setminus \Omega. \tag{4.6.21}$$

It follows from Theorem 4.3.2 that there is a constant $C > 0$ such that

$$\|w\|_{W^{s,2}(\mathbb{R}^N)} \leq C\|\eta\|_{L^2(\Omega)}. \tag{4.6.22}$$

Since $w \in W_0^{s,2}(\overline{\Omega})$, then using (4.6.18) we get that

$$\begin{aligned}
&\int_\Omega (u - u_n)(-\Delta)^s w \, dx \\
&= \frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{((u-u_n)(x) - (u-u_n)(y))(w(x) - w(y))}{|x-y|^{N+2s}} \, dxdy \\
&\quad - \int_{\mathbb{R}^N\setminus\Omega} (u - u_n)\mathcal{N}_s w \, dx \\
&= \mathcal{E}_n(u - u_n, w) - \int_{\mathbb{R}^N\setminus\Omega} (u - u_n)\mathcal{N}_s w \, dx \\
&= \int_{\mathbb{R}^N\setminus\Omega} w\mathcal{N}_s u \, dx - \int_{\mathbb{R}^N\setminus\Omega} (u - u_n)\mathcal{N}_s w \, dx \\
&= -\int_{\mathbb{R}^N\setminus\Omega} (u - u_n)\mathcal{N}_s w \, dx.
\end{aligned}$$

It follows from the preceding identity, (4.6.19) and (4.6.22) that there is a constant $C > 0$

such that

$$\left|\int_{\Omega}(u-u_n)(-\Delta)^s w\;dx\right| = \left|\int_{\mathbb{R}^N\setminus\Omega}(u-u_n)\mathcal{N}_s w\;dx\right|$$
$$\leq \|u-u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|\mathcal{N}_s w\|_{L^2(\mathbb{R}^N\setminus\Omega)}$$
$$\leq \frac{C}{n}\|u\|_{W^{s,2}(\mathbb{R}^N)}\|w\|_{W^{s,2}(\mathbb{R}^N)}$$
$$\leq \frac{C}{n}\|u\|_{W^{s,2}(\mathbb{R}^N)}\|\eta\|_{L^2(\Omega)}. \tag{4.6.23}$$

Using (4.6.20) and (4.6.23) we get that

$$\|u-u_n\|_{L^2(\Omega)} \leq \frac{C}{n}\|u\|_{W^{s,2}(\mathbb{R}^N)}. \tag{4.6.24}$$

Now the estimate (4.6.14) follows from (4.6.19) and (4.6.24). Observe that it follows from (4.6.14) that $u_n \to u$ in $L^2(\mathbb{R}^N)$ as $n \to \infty$ and this completes the proof of Part (a).

(b) Now let $z \in L^2(\mathbb{R}^N \setminus \Omega) \hookrightarrow L^2(\mathbb{R}^N \setminus \Omega, \mu)$. Notice that $\{u_n\}_{n\in\mathbb{N}}$ satisfies (4.6.16). Proceeding as in the proof of Theorem 4.6.2 we can deduce that there is a constant $C > 0$ (independent of n) such that

$$n\|u_n\|^2_{L^2(\mathbb{R}^N\setminus\Omega)} \leq C\mathcal{E}_n(u_n,u_n) \leq nC\|\kappa\|_{L^\infty(\mathbb{R}^N\setminus\Omega)}\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)},$$

and this implies that

$$\|u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)} \leq C\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}. \tag{4.6.25}$$

Now we proceed as in the proof of (4.6.24). As in (4.6.20) we have that

$$\|u_n\|_{L^2(\Omega)} = \sup_{\eta\in L^2(\Omega)} \frac{\left|\int_\Omega u_n\eta\ dx\right|}{\|\eta\|_{L^2(\Omega)}}. \tag{4.6.26}$$

Let $\eta \in L^2(\Omega)$ and $w \in W_0^{s,2}(\overline{\Omega})$ the weak solution of (4.6.21). Since $w \in W_0^{s,2}(\overline{\Omega})$, then using (4.6.18) we have that

$$\begin{aligned}
&\int_\Omega u_n(-\Delta)^s w\ dx\\
&=\frac{C_{N,s}}{2}\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u_n(x)-u_n(y))(w(x)-w(y))}{|x-y|^{N+2s}}\ dxdy - \int_{\mathbb{R}^N\setminus\Omega} u_n\mathcal{N}_s w\ dx\\
&=-\int_{\mathbb{R}^N\setminus\Omega} u_n\mathcal{N}_s w\ dx.
\end{aligned}$$

It follows from the preceding identity, (4.6.25) and (4.6.22) that there is a constant $C > 0$ such that

$$\begin{aligned}
\left|\int_\Omega u_n(-\Delta)^s w\ dx\right| = \left|\int_{\mathbb{R}^N\setminus\Omega} u_n\mathcal{N}_s w\ dx\right| &\le \|u_n\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|\mathcal{N}_s w\|_{L^2(\mathbb{R}^N\setminus\Omega)}\\
&\le C\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}\|w\|_{W^{s,2}(\mathbb{R}^N)}.
\end{aligned} \tag{4.6.27}$$

Using (4.6.25), (4.6.27) and (4.6.22) we get that there is a constant $C > 0$ (independent of n) such that

$$\|u_n\|_{L^2(\Omega)} \le C\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}. \tag{4.6.28}$$

Combining (4.6.25) and (4.6.28) we get that

$$\|u_n\|_{L^2(\mathbb{R}^N)} \le C\|z\|_{L^2(\mathbb{R}^N\setminus\Omega)}. \tag{4.6.29}$$

Hence, the sequence $\{u_n\}_{n\in\mathbb{N}}$ is bounded in $L^2(\mathbb{R}^N)$. Thus, after a subsequence, if necessary, we have that u_n converges weakly to some $\tilde{u}$ in $L^2(\mathbb{R}^N)$ as $n \to \infty$.

Using (4.6.16) we get that for every $v \in V := \{v \in W_0^{s,2}(\overline{\Omega}) :\ (-\Delta)^s v \in L^2(\Omega)\}$,

$$\frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u_n(x) - u_n(y))(v(x) - v(y))}{|x-y|^{N+2s}}\, dxdy = 0. \tag{4.6.30}$$

Using the integration by parts formula (4.2.5) we can deduce that

$$\begin{aligned}\frac{C_{N,s}}{2} \int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} &\frac{(u_n(x) - u_n(y))(v(x) - v(y))}{|x-y|^{N+2s}}\, dxdy \\ &= \int_\Omega u_n(-\Delta)^s v\, dx + \int_{\mathbb{R}^N\setminus\Omega} u_n \mathcal{N}_s v\, dx,\end{aligned} \tag{4.6.31}$$

for every $v \in V$. Combining (4.6.30) and (4.6.31) we get that the identity

$$\int_\Omega u_n(-\Delta)^s v\, dx + \int_{\mathbb{R}^N\setminus\Omega} u_n \mathcal{N}_s v\, dx = 0, \tag{4.6.32}$$

holds for every $v \in V$. Passing to the limit in (4.6.32) as $n \to \infty$, we obtain that

$$\int_\Omega \tilde{u}(-\Delta)^s v\, dx + \int_{\mathbb{R}^N\setminus\Omega} \tilde{u} \mathcal{N}_s v\, dx = 0,$$

for every $v \in V$. We have shown (4.6.15) and the proof is finished. □

Toward this end, for $\xi \geq 0$ we introduce the regularized fractional Dirichlet control problem:

$$\min_{u\in U_R, z\in Z_R} J(u) + \frac{\xi}{2}\|z\|^2_{L^2(\mathbb{R}^N\setminus\Omega)}, \tag{4.6.33a}$$

subject to the regularized exterior value problem (Robin problem): Find $u_n \in U_R$ solving

$$\begin{cases} (-\Delta)^s u & = 0 \quad \text{in } \Omega \\ \mathcal{N}_s u + n\kappa u & = n\kappa z \quad \text{in } \mathbb{R}^N \setminus \Omega, \end{cases} \tag{4.6.33b}$$

and the control constraints

$$z \in Z_{ad,R}. \tag{4.6.33c}$$

Here $Z_R := L^2(\mathbb{R}^N \setminus \Omega)$, $Z_{ad,R}$ is a closed and convex subset of Z_R and $U_R := W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)$. We again remark that (4.6.33) is nothing but the fractional Robin exterior control problem.

Theorem 4.6.5 (**Approximation of the Dirichlet Control Problem).** The regularized control problem (4.6.33) admits a minimizer $(z_n, u(z_n)) \in Z_{ad,R} \times (W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega))$. Let $Z_R = W^{s,2}(\mathbb{R}^N \setminus \Omega)$ and $Z_{ad,R} \subset Z_R$ be bounded. Then for any sequence $\{n_\ell\}_{\ell=1}^\infty$ with $n_\ell \to \infty$, there exists a subsequence still denoted by $\{n_\ell\}_{\ell=1}^\infty$ such that $z_{n_\ell} \rightharpoonup \tilde{z}$ in $W^{s,2}(\mathbb{R}^N \setminus \Omega)$, $u(z_{n_\ell}) \to \tilde{u}$ in $L^2(\mathbb{R}^N)$ as $n_\ell \to \infty$ and $(\tilde{z}, \tilde{u})$ solves the Dirichlet control problem (4.1.3) with $Z_{ad,D}$ replaced by $Z_{ad,R}$.

Proof. Since the regularized control problem (4.6.33) is nothing but the Robin control problem therefore the existence of minimizers follows by directly using Theorem 4.5.1. Following the proof of Theorem 4.5.1 and using the fact that $Z_{ad,R}$ is a bounded subset of the reflexive Banach space $W^{s,2}(\mathbb{R}^N \setminus \Omega)$, after a subsequence, if necessary, we have that $z_{n_\ell} \rightharpoonup \tilde{z}$ in $W^{s,2}(\mathbb{R}^N \setminus \Omega)$ as $n_\ell \to \infty$. Now since $Z_{ad,R}$ is closed and convex, then it is weakly closed. Thus $\tilde{z} \in Z_{ad,R}$.

Following the proof of Theorem 4.6.3(a) we can deduce that there exists a subsequence $\{u_{n_\ell}\}$ such that $u_{n_\ell} \to \tilde{u}$ in $L^2(\mathbb{R}^N)$ as $n_\ell \to \infty$ and $\tilde{u} \in W^{s,2}(\mathbb{R}^N)$. Combining this convergence with the aforementioned convergence of z_{n_ℓ} we can conclude that $(\tilde{z}, \tilde{u}) \in$

$Z_{ad,R} \times W^{s,2}(\mathbb{R}^N)$ solves the Dirichlet exterior value problem (4.1.3b).

It then remains to show that $(\tilde{z}, \tilde{u})$ is a minimizer of (4.1.3). Let (z', u') be any minimizer of (4.1.3). Let us consider the regularized state equation (4.6.33b) but with boundary datum z'. We denote the solution of the resulting state equation by u'_{n_ℓ}. By using the same limiting argument as above we can select a subsequence such that $u'_{n_\ell} \to u'$ in $L^2(\mathbb{R}^N)$ as $n_\ell \to \infty$. Letting $j(z,u) := J(u) + \frac{\xi}{2}\|z\|^2_{L^2(\mathbb{R}^N\setminus\Omega)}$, it then follows that

$$j(z', u') \leq j(\tilde{z}, \tilde{u}) \leq \liminf_{n_\ell\to\infty} j(z_{n_\ell}, u_{n_\ell}) \leq \liminf_{n_\ell\to\infty} j(z', u'_{n_\ell}) = j(z', u'),$$

where the second inequality is due to the weak-lower semicontinuity of J. The third inequality is due to the fact that $\{(z_{n_\ell}, u_{n_\ell})\}$ is a sequence of minimizers for (4.6.33). The proof is finished. □

We conclude this section by writing the stationarity system corresponding to (4.6.33): Find $(z, u, p) \in Z_{ad,R} \times (W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)) \times (W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega))$ such that

$$\begin{cases} \mathcal{E}(u,v) = \displaystyle\int_{\mathbb{R}^N\setminus\Omega} n\kappa z v \, dx, \\ \mathcal{E}(w,p) = \displaystyle\int_{\Omega} J'(u) w \, dx, \\ \displaystyle\int_{\mathbb{R}^N\setminus\Omega} (n\kappa p + \xi z)(\widetilde{z} - z) \, dx \geq 0, \end{cases} \tag{4.6.34}$$

for all $(\widetilde{z}, v, w) \in Z_{ad,R} \times (W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega)) \times (W^{s,2}_{\Omega,\kappa} \cap L^2(\mathbb{R}^N \setminus \Omega))$.

4.7 Numerical Approximations

The purpose of this section is to introduce numerical approximations of the problems we have considered so far. We emphasize that the fractional PDEs are intrinsically expensive, since the underlying coefficient matrices require approximation of integrals with singular kernels and the resulting coefficient matrices are dense. In addition, the fractional control problems can be prohibitively expensive since they not only require solving the nonlocal state equation but also the nonlocal adjoint equation and in case of Dirichlet control problem one needs to approximate the nonlocal normal derivative for the adjoint variable to evaluate the control variable (4.4.6). The presented approach is a first of its kind to numerically solve the fractional Robin problem and to approximate the Dirichlet problem by the Robin problem. Further details on the numerical analysis of the underlying PDEs, the control problem and solvers for the fractional PDEs will be part of forthcoming work.

The rest of the section is organized as follows: In Subsection 4.7.1 we begin with a finite element approximation of the Robin problem (4.6.1) which is the same as the regularized Dirichlet problem. We approximate the Dirichlet problem using the Robin problem. In Subsection 4.7.2 we introduce an external source identification problem where we clearly see the difference between the nonlocal case and the classical case ($s \sim 1$). Finally, Subsection 4.7.3 is devoted to the optimal control problems.

4.7.1 Approximation of a Nonhomogeneous Dirichlet Problem via a Robin Problem

In view of Theorem 4.6.3 we can approximate the Dirichlet problem with the help of the Robin (regularized Dirichlet) problem (4.6.1). Therefore we begin by introducing a discrete scheme for the Robin problem. Let $\widetilde{\Omega}$ be a bounded open set that contains Ω, the support of the control/source z and the support of κ. We consider a conforming simplicial triangulation of Ω and $\widetilde{\Omega} \setminus \Omega$ such that the resulting partition remains admissible. We shall assume that the support of z and κ is contained in $\widetilde{\Omega} \setminus \Omega$. We let our finite element space $\mathbb{V}_h$ (on $\widetilde{\Omega}$) to be a set of continuous piecewise linear functions. Then the discrete (weak) version of

(4.6.33b) with nonzero right-hand-side is given as follows: Find $u_h \in \mathbb{V}_h$ such that

$$\begin{aligned}\int\int_{\mathbb{R}^{2N}\setminus(\mathbb{R}^N\setminus\Omega)^2} \frac{(u_h(x)-u_h(y))(v(x)-v(y))}{|x-y|^{N+2s}}\,dxdy + \int_{\widetilde{\Omega}\setminus\Omega} n\kappa u_h v\,dx \\ = \langle f, v\rangle_{(W^{s,2}_{\Omega,\kappa}\cap L^2(\mathbb{R}^N\setminus\Omega))^\star, W^{s,2}_{\Omega,\kappa}\cap L^2(\mathbb{R}^N\setminus\Omega)} + \int_{\widetilde{\Omega}\setminus\Omega} n\kappa z v\,dx \quad \forall v \in \mathbb{V}_h.\end{aligned} \tag{4.7.1}$$

We approximate the double integral over $\mathbb{R}^{2N} \setminus (\mathbb{R}^N \setminus \Omega)^2$ by using the approach from [1, 2]. The remaining integrals are computed using numerical quadrature which is accurate for polynomials of degree less than and equal to 4. All implementations are carried out in Matlab and the discrete system of equations corresponding to the state and adjoint equations are solved using direct solvers. Note that iterative solvers for the fractional Robin problem are part of our future work.

We next consider an example that has been taken from [3]. Let $\Omega = B_0(1/2) \subset \mathbb{R}^2$. Our goal is to find u solving

$$\begin{cases}(-\Delta)^s u = 2 & \text{in } \Omega,\\ u(\cdot) = \frac{2^{-2s}}{\Gamma(1+s)^2}\left(1-|\cdot|^2\right)_+^s & \text{in } \mathbb{R}^N\setminus\Omega.\end{cases}$$

The exact solution in this case is given by

$$u(x) = u_1(x) + u_2(x) = \frac{2^{-2s}}{\Gamma(1+s)^2}\left(\left(1-|x|^2\right)_+^s + \left(\frac{1}{4}-|x|^2\right)_+^s\right),$$

where u_1 and u_2 solve the problems

$$\begin{cases}(-\Delta)^s u_1 = 1 & \text{in } \Omega,\\ u_1 = \frac{2^{-2s}}{\Gamma(1+s)^2}\left(1-|\cdot|^2\right)_+^s & \text{in } \mathbb{R}^N\setminus\Omega,\end{cases} \qquad \begin{cases}(-\Delta)^s u_2 = 1 & \text{in } \Omega,\\ u_2 = 0 & \text{in } \mathbb{R}^N\setminus\Omega.\end{cases} \tag{4.7.2}$$

We let $\widetilde{\Omega} = B_0(3/2)$. We next approximate (4.7.2) using (4.7.1) and we set $\kappa = 1$ on its support. At first we fix $s = 0.5$ and the Degrees of Freedom (DoFs) to be DoFs $= 2920$. For this configuration, we study the $L^2(\Omega)$ error $\|u - u_h\|_{L^2(\Omega)}$ with respect to n in Figure 4.2 (left). As expected, from Theorem 4.6.3(a) we observe an approximation rate of $1/n$.

Next for a fixed $s = 0.5$, we check the stability of our scheme with respect to n as we refine the mesh. We have plotted the L^2-error as we refine the mesh (equivalently increase DOFs) for $n = 1e2$, $1e3$, $1e4$, $1e5$. We notice that the error remains stable with respect to n and we observe the following convergence rate with respect to the DoFs:

$$\|u - u_h\|_{L^2(\Omega)} \approx (\text{DoFs})^{-\frac{1}{2}}.$$

In the right panel we have shown the L^2-error for a fixed $n = 1e5$ but for various $s = 0.2$, 0.4, 0.6, 0.8. When $0 < s < \frac{1}{2}$ we have observed a rate of $(\text{DoFs})^{-\frac{1}{2}\left(s+\frac{1}{2}\right)}$ and for $\frac{1}{2} \leq s < 1$ we observe a convergence rate of $(\text{DoFs})^{-\frac{1}{2}}$.

4.7.2 External Source Identification Problem

We next consider an inverse problem to identify a source that is located outside the observation domain Ω. The optimality system is as given in (4.6.34) where we have approximated the Dirichlet problem by the Robin problem. We use the continuous piecewise linear finite element discretization for all the optimization variables: state (u), control (z) and adjoint (p). We choose our objective function as

$$j(u, z) = J(u) + \frac{\xi}{2}\|z\|^2_{L^2(\mathbb{R}^N \setminus \Omega)}, \quad \text{with} \quad J(u) := \frac{1}{2}\|u - u_d\|^2_{L^2(\Omega)},$$

and we let $Z_{ad,R} := \{z \in L^2(\mathbb{R}^N \setminus \Omega) \ : \ z \geq 0, \text{ a.e. in } \widehat{\Omega}\}$ where $\widehat{\Omega}$ is the support set of the control z that is contained in $\widetilde{\Omega} \setminus \Omega$. Moreover $u_d : L^2(\Omega) \to \mathbb{R}$ is the given data (observations). All the optimization problems below are solved using the projected-BFGS

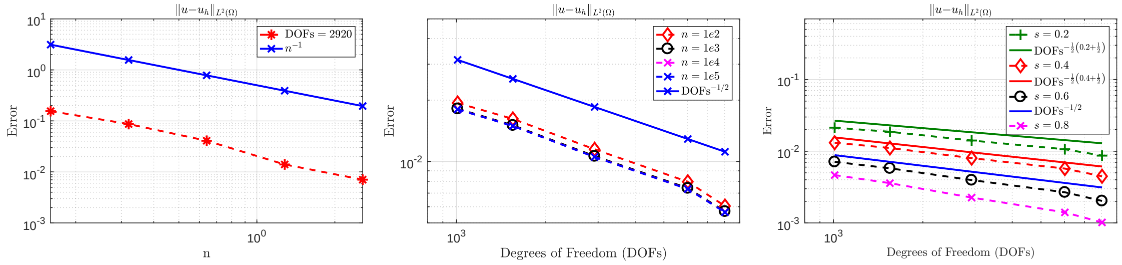

Figure 4.2: Left panel: Let $s = 0.5$ and DoFs $= 2920$ be fixed. We let $\kappa = 1$ and consider the L^2-error between the actual solution u to the Dirichlet problem and its approximation u_h which solves the Robin problem. We have plotted the error with respect to n. The solid line denotes a reference line and the actual error. We observe a rate of $1/n$ which confirms our theoretical result (4.6.14). Middle panel: Let $s = 0.5$ be fixed. For each $n = 1e2, 1e3, 1e4, 1e5$ we have plotted the L^2-error with respect to the degrees of freedom (DOFs) as we refine the mesh. Notice that the error is stable with respect to n. Moreover, the observed rate of convergence is $(\mathrm{DoFs})^{-\frac{1}{2}}$ and is independent of n. Right panel: Let $n = 1e5$ be fixed. We plot the L^2-error with respect to the DOFs for various values of s. The observed convergence rate is $(\mathrm{DoFs})^{-\frac{1}{2}\left(s+\frac{1}{2}\right)}$ for $0 < s < \frac{1}{2}$ and the observed rate is $(\mathrm{DoFs})^{-\frac{1}{2}}$ for $\frac{1}{2} \leq s < 1$.

method with Armijo line search.

Our computational setup is shown in Figure 4.3. The centered square region is $\Omega = [-0.4, 0.4]^2$ and the region inside the outermost ring is $\widetilde{\Omega} = B_0(3/2)$. The smaller square inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the support of the source/control. The right panel in Figure 4.3 shows a finite element mesh with DoFs $= 6103$.

We define u_d as follows. For $z = 1$, we first solve the state equation for $\tilde{u}$ (first equation in (4.6.34)). We then add a normally distributed random noise with mean zero and standard deviation 0.02 to $\tilde{u}$. We call the resulting expression as u_d. Furthermore, we set $\kappa = 1$, and $n = 1e5$.

Our goal is then to identify the source $\bar{z}_h$. In Figure 4.4, we first show the behavior of optimal $\bar{z}_h$ for different values of the regularization parameter $\xi = 1e-1$ (4), $1e-2$ (4), $1e-4$ (4), $1e-8$ (2), $1e-10$ (2). The numbers in the bracket denote the total number of iterations that the BFGS has taken to achieve a stopping tolerance (for the projected

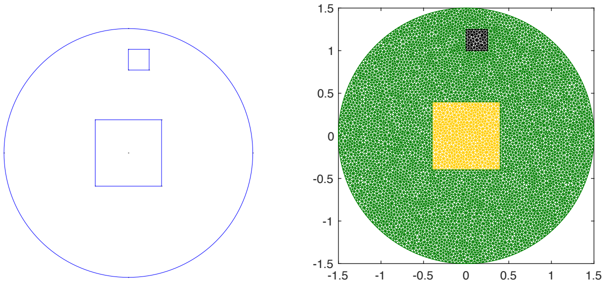

Figure 4.3: Left: computational domain where the inner square is Ω, the region inside the outer circle is $\widetilde{\Omega}$ and the outer square inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the region where the source/control is supported. Right: A finite element mesh.

gradient) of $1e-7$. Notice that the Armijo line search has remained inactive in these cases. As expected the larger is the value of ξ, the smaller is the magnitude of $\bar{z}_h$, and this behavior saturates at $\xi = 1e-8$.

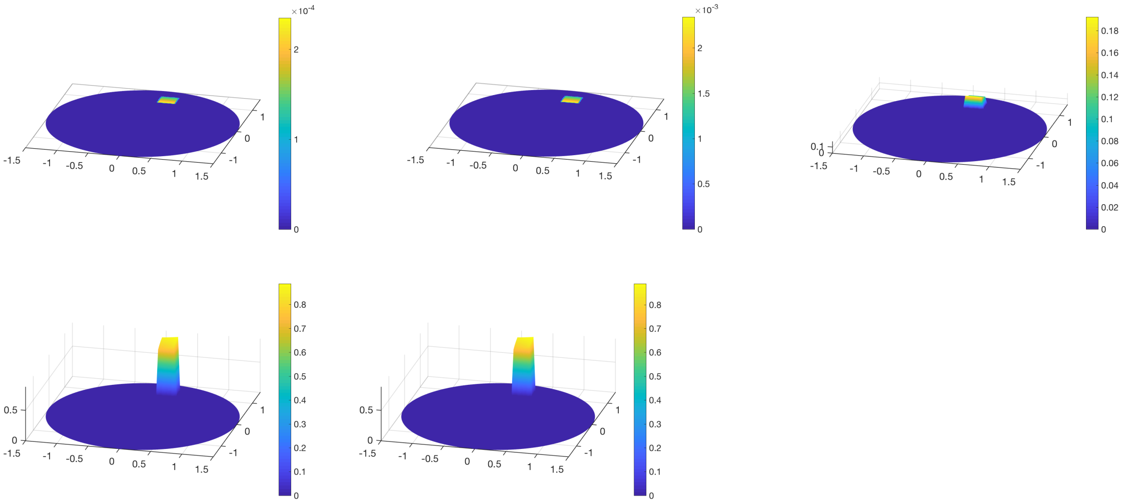

Figure 4.4: External source identification problem. The panels show the behavior of $\bar{z}_h$ with respect to the regularization parameter ξ: top row from left to right $\xi = 1e-1, 1e-2, 1e-4$; bottom row from left to right: $\xi = 1e-8, 1e-10$. As it is expected, larger is ξ, smaller is the magnitude of $\bar{z}_h$, but this behavior saturates at $\xi = 1e-8$.

Next, for a fixed $\xi = 1e-8$, Figure 4.5 shows the optimal $\bar{z}_h$ for $s = 0.1$ (4), 0.6 (2), 0.7 (2), 0.8 (2), 0.9 (2). The numbers in the bracket again denote the total number of iterations that the BFGS has taken to achieve a stopping tolerance (for the projected gradient) of $1e-7$. Notice that the Armijo line search has remained inactive in these cases. We notice that for large s, $\bar{z}_h \equiv 0$. This is expected as larger the s is, the more close we are to the classical Poisson problem case and we know that we cannot impose the external condition in that case.

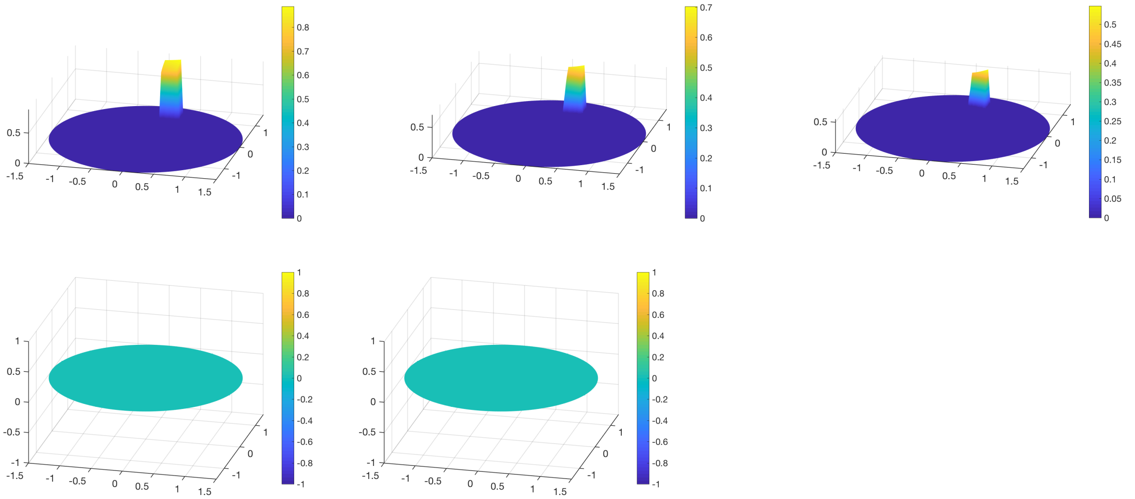

Figure 4.5: The panels show the behavior of $\bar{z}_h$ as we vary the exponent s. Top row from left to right: $s = 0.1, 0.6, 0.7$. Bottom row from left to right: $s = 0.8, 0.9$. For smaller values of s, the recovery of $\bar{z}_h$ is quite remarkable. However, for larger values of s, $\bar{z}_h \equiv 0$ as expected, the behavior of $\bar{u}_h$ for large s is close to the classical Poisson problem which does not allow external sources.

4.7.3 Dirichlet Control Problem

We next consider two Dirichlet control problems. The setup is similar to Subsection 4.7.2 except now we set $u_d \equiv 1$.

Example 4.7.1. The computational setup for the first example is shown in Figure 4.6. Let

$\Omega = B_0(1/2)$ (the region insider the innermost ring) and the region inside the outermost ring is $\widetilde{\Omega} = B_0(3/2)$. The annulus inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the support of the control. The right panel in Figure 4.6 shows a finite element mesh with DoFs = 6069.

In Figures 4.7 and 4.8 we have shown the optimization results for $s = 0.2$ (14) and $s = 0.8$ (4), respectively. Here again, the numbers in the bracket denote the total number of iterations that the BFGS has taken to achieve a stopping tolerance (for the projected gradient) of $1e-7$. Notice that the Armijo line search has remained inactive in these cases. The top row shows the desired state u_d (left) and the optimal state $\bar{u}_h$ (right). The bottom row shows the optimal control $\bar{z}_h$ (left) and the optimal adjoint variable $\bar{p}_h$ (right). We notice that in both cases we can approximate the desired state to a high accuracy but the approximation is slightly better for smaller s, especially close to the boundary. This is to be expected as for large values of s the regularity of the adjoint variable deteriorates significantly (cf. Theorem 4.4.4).

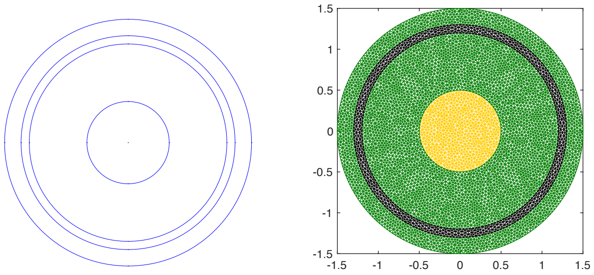

Figure 4.6: Left: computational domain where the inner circle is Ω, the region inside the outer circle is $\widetilde{\Omega}$, and the annulus inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the region where the control is supported. Right: A finite element mesh.

Example 4.7.2. The computational setup for our final example is shown in Figure 4.9. The M-shape region is Ω and the region inside the outermost ring is $\widetilde{\Omega} = B_0(0.6)$. The smaller region inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the support of the control. The right panel in Figure 4.6 shows a finite element mesh with DoFs = 4462.

In Figure 4.10 we have shown the optimization results for $s = 0.8$ (370). Again, the

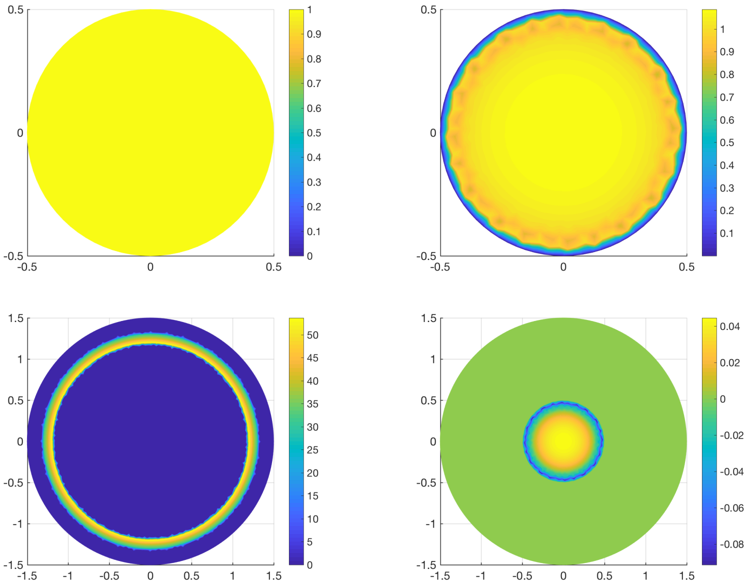

Figure 4.7: Example 1, $s = 0.2$: Top row: Left - Desired state u_d; Right - Optimal state $\bar{u}_h$. Bottom row: Left - Optimal control $\bar{z}_h$, Right - Optimal adjoint $\bar{p}_h$.

number in the bracket denotes the total number of iterations that the BFGS has taken to achieve a stopping tolerance (for the projected gradient) of $1e-7$. Notice for this example, during most of the iterations, the Armijo line search has remained inactive but it got activated during a few number of iterations and has remained active for up to 10 steps. The top row shows the desired state u_d (left) and the optimal state $\bar{u}_h$ (right). The bottom row shows the optimal control $\bar{z}_h$ (left) and the optimal adjoint variable $\bar{p}_h$ (right). Even though the control is applied in an extremely small region we can still match the desired state in certain parts of Ω.

4.8 Discussion

There are many phenomenon in nature which can be modeled using optimization problems constrained by PDEs. These models allow identification of source, or an enforcement of control using the experimental data. Classical models allow placement of the control or

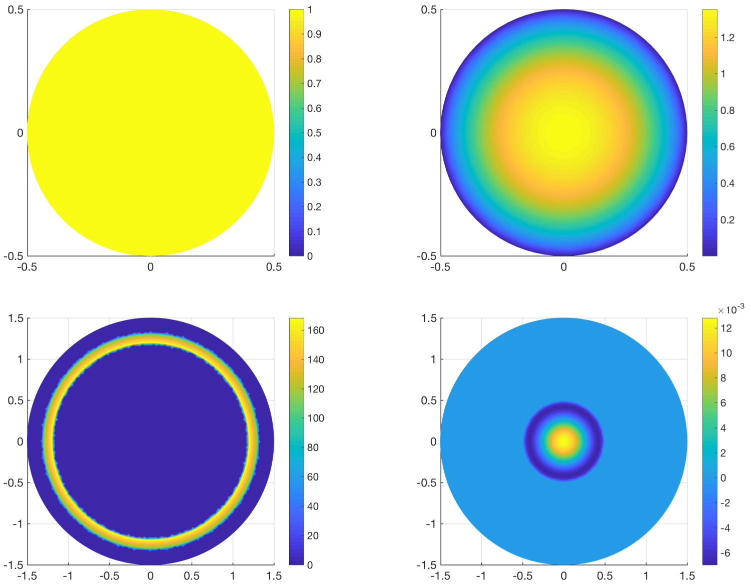

Figure 4.8: Example 1, $s = 0.8$: Top row: Left - Desired state u_d; Right - Optimal state $\bar{u}_h$. Bottom row: Left - Optimal control $\bar{z}_h$, Right - Optimal adjoint $\bar{p}_h$.

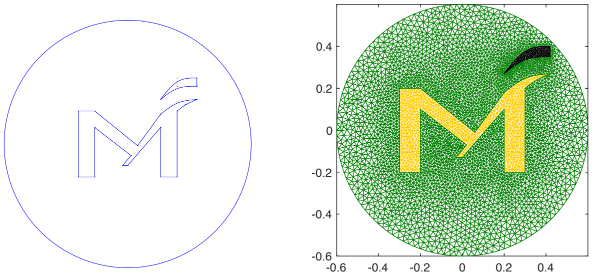

Figure 4.9: Left: computational domain where the M-shaped region is Ω, the region inside the outer circle is $\widetilde{\Omega}$ and the region inside $\widetilde{\Omega} \setminus \Omega$ is $\widehat{\Omega}$ which is the region where control is supported. Right: A finite element mesh.

source only on the boundary or inside the observation domain. The question that is of particular interest to us is the placement of source/control outside and away from the observation domain. Classical models cannot be used for this purpose as they stand as they involve local operators, which have a limited reach. In the current work, we have

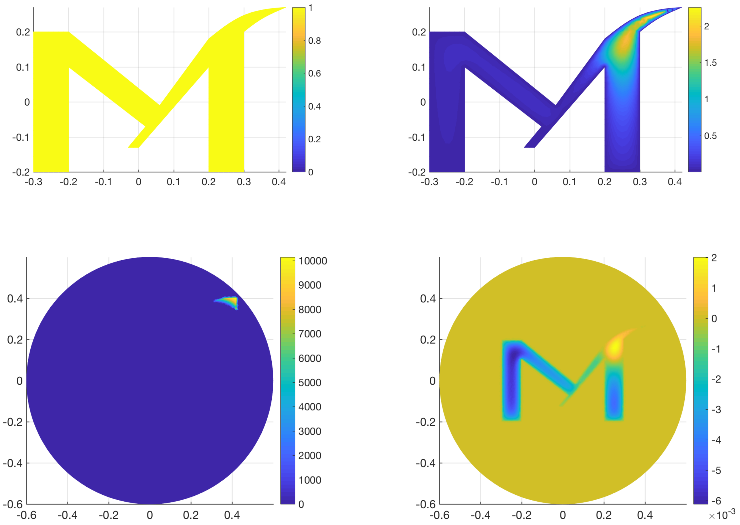

Figure 4.10: Example 3, $s = 0.8$: Top row: Left - Desired state u_d; Right - Optimal state $\bar{u}_h$. Bottom row: Left - Optimal control $\bar{z}_h$, Right - Optimal adjoint $\bar{p}_h$.

introduced a new model of inverse problems constrained by elliptic fractional PDEs, which involve nonlocal operators. This gives us the flexibility to incorporate a source/control which is external (and disjoint) to the observation domain. In this context, we have introduced the Dirichlet, Robin and Neumann exterior control problems. In particular, we have studied the Dirichlet and Robin exterior control problems. The case of Neumann exterior control problem follows very similarly. The fractional elliptic equation that we have considered is the fractional Poisson equation, which involves the fractional Laplacian operator.

For this class of inverse problems we have introduced the notion of very-weak solutions to the Dirichlet exterior problem and weak solutions of the Robin exterior problem. These notions can be used for the optimal control problem. Furthermore, approximating the weak solution of the Dirichlet exterior value problem requires approximation of the nonlocal normal derivative, which is quite challenging. We have introduced a way around this by considering a suitable Robin exterior control problem to approximate the weak solutions of nonhomogeneous Dirichlet exterior value problem. Consequently, we have approximated the solution to the Dirichlet exterior control problem by a Robin exterior control problem, along with an experimental rate of convergence.

We have shown several numerical experiments where we have discussed both external source identification as well as external control problems. In particular, we have established that for smaller values of the fractional exponent, we are better able to locate the external source. Similarly, a smaller exponent also allows for a better exterior control. The examples that we have considered in this work have been synthetically generated. Application of this work to real life applications with actual experimental data would be part of our future work.

Chapter 5: Conclusions and Discussion

In this study, we have motivated the use of nonlocal operators in the context of 3 different problems connected by nonlocal operators. The two nonlocal operators we have considered are the fractional Laplacian and the Caputo fractional time derivative.

At first, in Chapter 2 we have considered an inverse problem from imaging science, where we have proposed the use of fractional Laplacian as a regularizer to improve the reconstruction quality. We have compared this regularization with the state of the art total variation regularization. The inevitable question that follows with the use of regularization is finding the strength of regularization, for which there is no state of the art method. To address this challenge, we have developed a deep learning based algorithmic framework of a Bilevel Optimization Neural Network, BONNet, to learn the optimal regularization strength. We have considered tomographic reconstruction as our model problem, and shown an improvement in the reconstruction quality, specially due to the fractional Laplacian regularization, where the optimal regularization strength is determined via BONNet. The fractional Laplacian regularization, along with the regularization strength finding BONNet algorithm, has turned out to be highly efficient in reconstructing internal features of the sample, specially for limited and noise-filled data. We also remark that this is the first instance where a neural network has been used to determine the fractional exponent of the fractional Laplacian, which contributes to the strength of this regularization. The details of study can be found in Chapter 2, as well as [8].

Next, in Chapter 3 we have further explored the avenue of enriching deep neural networks via nonlocal operators. We have developed a novel continuous model and stable discretization of DNNs that incorporate history. We have accomplished this via an optimal control formulation of a DNN bestowed with Caputo fractional time derivative. We have presented the algorithmic framework of this network, which we call fractional-DNN. We expect keeping track of history in this manner can alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse and reduce the number of unknown parameters. We have considered various datasets and shown an improvement in the classification accuracy via fractional-DNN compared to the standard Residual Neural Network. We have also numerically demonstrated the improvement in the vanishing gradient problem, as well as flow of features as a dynamical system. The benefit of fractional-DNN is that it has a rigorous mathematical foundation and framework which establishes a deeper understanding of DNNs with memory. This enhances their applicability to scientific and engineering applications. The details of study can be found in Chapter 3, as well as [9].

Lastly, in Chapter 4 we have presented a new class of inverse problems in which the control/source is located external to the observation domain. In contrast, classical diffusion models only allow control or source placement either inside the observation domain or on its boundary $\partial\Omega$. This can be prohibitive for high consequence applications such as magnetic drug targeting where the control (magnets) are away from the boundary [13, 14]. We achieve this feat via the nonlocality of the fractional Laplacian operator. We have shown via numerical examples that a smaller exponent of the fractional Laplacian leads to a better detection of the external source, as well as a better external control. In contrast, we also observe that larger values of the fractional exponent, which approach the classical models, cannot facilitate the external location of the control and source. The details of study can be found in Chapter 4, as well as [10].

Thus forth the purpose of this study has been to introduce novel ideas of using nonlocal operators in inverse problems as well as deep learning. We have presented two algorithmic frameworks, namely BONNet and fractional-DNN, as well a mathematically rigorous model of handling external control or source. We remark that our work has already been adapted for further scientific pursuits and investigations. Consider, e.g. our work on the fractional Laplacian regularization. One particular case of our work, dealing with image denoising in a bilevel optimization framework, has been expanded in [28], where the authors assume access to the true solution (i.e. the denoised image). In contrast, only the training phase of our algorithm needs access to the true solution, which, in fact, can be synthetically generated in the absence of true solution. Our BONNet architecture can thus be used for image reconstruction/denoising even in the absence of true solution. Another direct adaptation of our work can be seen in [19], where the authors have considered the external optimal control model where the constraint equation is a parabolic fractional PDE instead of an elliptic fractional PDE.

Moving forward, there are a number of paths for this research to expand further. First and foremost is the optimization of the algorithmic frameworks for large scale applications. Next, to enhance further applicability of these algorithmic frameworks, they need to be adapted to open access platforms by incorporating them into libraries for open source languages, e.g. Python, to facilitate a ready usage of the codes. Furthermore, for the fractional-DNN architecture we have primarily considered classification problems which fall under the category of supervised machine learning. We would be interested in expanding the current work to adapt to the semi-supervised as well as unsupervised learning. This will open doors to study a much wider class of important and challenging problems in scientific and technological applications which can be enriched via nonlocal operators, e.g. adversarial learning. Lastly, our work on the external optimal control can be expanded via applications to real-world problems, e.g. those stemming from Geophysical Electromagnetic experiments.

Bibliography

Bibliography

[1] G. Acosta, F.M. Bersetche, and J.P. Borthagaray. A short FE implementation for a 2d homogeneous Dirichlet problem of a fractional Laplacian. *Computers & Mathematics with Applications*, 74(4):784–816, 2017.

[2] G. Acosta and J.P. Borthagaray. A fractional Laplace equation: regularity of solutions and finite element approximations. *SIAM J. Numer. Anal.*, 55(2):472–495, 2017.

[3] G. Acosta, J.P. Borthagaray, and N. Heuer. Finite element approximations of the nonhomogeneous fractional Dirichlet problem. *arXiv preprint arXiv:1709.06592*, 2017.

[4] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems.* Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, New York, 2000.

[5] D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[6] H. Antil and S. Bartels. Spectral Approximation of Fractional PDEs in Image Processing and Phase Field Modeling. *Comput. Methods Appl. Math.*, 17(4):661–678, 2017.

[7] H. Antil, T. Berry, and J. Harlim. Fractional diffusion maps. *arXiv preprint arXiv:1810.03952*, 2018.

[8] H. Antil, Z. Di, and R. Khatri. Bilevel optimization, deep learning and fractional laplacian regularization with applications in tomography. *Inverse Problems*, 2020.

[9] H. Antil, R. Khatri, R. Löhner, and D. Verma. Fractional deep neural network via constrained optimization. *arXiv preprint arXiv:2004.00719*, 2020.

[10] H. Antil, R. Khatri, and M. Warma. External optimal control of nonlocal PDEs. *Inverse Problems*, 35(8):084003, 35, 2019.

[11] H. Antil, D.P. Kouri, M.D. Lacasse, and D. Ridzal (eds.). *Frontiers in PDE-constrained optimization.* The IMA Volumes in Mathematics and its Applications. Springer New York, 2018.

[12] H. Antil, C. Lizama, R. Ponce, and M. Warma. Convergence of solutions of discrete semi-linear space-time fractional evolution equations. *arXiv preprint arXiv:1910.07358*, 2019.

[13] H. Antil, R.H. Nochetto, and P. Venegas. Controlling the Kelvin force: basic strategies and applications to magnetic drug targeting. *Optim. Eng.*, 19(3):559–589, 2018.

[14] H. Antil, R.H. Nochetto, and P. Venegas. Optimizing the Kelvin force in a moving target subdomain. *Math. Models Methods Appl. Sci.*, 28(1):95–130, 2018.

[15] H. Antil, E. Otárola, and A.J. Salgado. A space-time fractional optimal control problem: analysis and discretization. *SIAM J. Control Optim.*, 54(3):1295–1328, 2016.

[16] H. Antil, J. Pfefferer, and S. Rogovs. Fractional operators with inhomogeneous boundary conditions: analysis, control, and discretization. *Commun. Math. Sci.*, 16(5):1395–1426, 2018.

[17] H. Antil, J. Pfefferer, and S. Rogovs. Fractional operators with inhomogeneous boundary conditions: Analysis, control, and discretization. *Communications in Mathematical Sciences (CMS)*, 16(5):1395–1426, 2018.

[18] H. Antil and C.N. Rautenberg. Sobolev spaces with non-Muckenhoupt weights, fractional elliptic operators, and applications. *SIAM J. Math. Anal.*, 51(3):2479–2503, 2019.

[19] H. Antil, D. Verma, and M. Warma. External optimal control of fractional parabolic PDEs. *ESAIM Control Optim. Calc. Var.*, 26, 2020.

[20] H. Antil and M. Warma. Optimal control of the coefficient for fractional p-Laplace equation: Approximation and convergence. *RIMS Kôkyûroku*, 2090:102–116, 2018.

[21] H. Antil and M. Warma. Optimal control of fractional semilinear PDEs. *To appear: Control, Optimisation and Calculus of Variations (ESAIM: COCV)*, 2019.

[22] H. Antil and M. Warma. Optimal control of the coefficient for regional fractional p-Laplace equations: Approximation and convergence. *Math. Control Relat. Fields.*, 9(1):1–38, 2019.

[23] T. Apel, S. Nicaise, and J. Pfefferer. Discretization of the Poisson equation with non-smooth data and emphasis on non-convex domains. *Numerical Methods for Partial Differential Equations*, 32(5):1433–1454, 2016.

[24] T. Apel, S. Nicaise, and J. Pfefferer. Adapted numerical methods for the Poisson equation with L^2 boundary data in nonconvex domains. *SIAM Journal on Numerical Analysis*, 55(4):1937–1957, 2017.

[25] H. Attouch, G. Buttazzo, and G. Michaille. *Variational analysis in Sobolev and BV spaces*, volume 17 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, second edition, 2014. Applications to PDEs and optimization.

[26] A. P. Austin, Z. Di, S. Leyffer, and S. M. Wild. Simultaneous sensing error recovery and tomographic inversion using an optimization-based approach. *SIAM Journal on Scientific Computing*, 41(3):B497–B521, 2019.

[27] S. Bartels and M. Milicevic. Alternating direction method of multipliers with variable step sizes. *arXiv preprint arXiv:1704.06069*, 2017.

[28] S. Bartels and N. Weber. Parameter learning and fractional differential operators: application in image regularization and decomposition. *arXiv preprint arXiv:2001.03394*, 2020.

[29] Y. Bengio. *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[30] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157,166, 1994-03.

[31] M. Benning, E. Celledoni, M. Ehrhardt, B. Owren, and C.-B. Schnlieb. Deep learning as optimal control problems: Models and numerical methods. *Journal of Computational Dynamics*, 6:171–198, 01 2019.

[32] M. Berggren. Approximations of very weak solutions to boundary-value problems. *SIAM J. Numer. Anal.*, 42(2):860–877 (electronic), 2004.

[33] U. Biccari, M. Warma, and E. Zuazua. Addendum: Local elliptic regularity for the Dirichlet fractional Laplacian. *Adv. Nonlinear Stud.*, 17(4):837–839, 2017.

[34] U. Biccari, M. Warma, and E. Zuazua. Local elliptic regularity for the Dirichlet fractional Laplacian. *Adv. Nonlinear Stud.*, 17(2):387–409, 2017.

[35] B. Bischke, P. Bhardwaj, A. Gautam, P. Helber, D. Borth, and A. Dengel. Detection of flooding events in social multimedia and satellite imagery using deep neural networks. In *Working Notes Proceedings of the MediaEval 2017. MediaEval Benchmark, September 13-15, Dublin, Ireland.* MediaEval, 2017.

[36] C. Bjorland, L. Caffarelli, and A. Figalli. Nonlocal tug-of-war and the infinity fractional Laplacian. *Comm. Pure Appl. Math.*, 65(3):337–380, 2012.

[37] L. Bottou, F.E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60(2):223–311, 2018.

[38] S. Bougleux, A. Elmoataz, and M. Melkemi. Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *International Journal of Computer Vision*, 84(2):220–236, Aug 2009.

[39] L. Brasco, E. Parini, and M. Squassina. Stability of variational eigenvalues for the fractional p-Laplacian. *Discrete Contin. Dyn. Syst.*, 36(4):1813–1845, 2016.

[40] T. Brown, S. Du, H. Eruslu, and F.-J. Sayas. Analysis of models for viscoelastic wave propagation. *arXiv preprint arXiv:1802.00825*, 2018.

[41] A. Bueno-Orovio, D. Kay, V. Grau, B. Rodriguez, and K. Burrage. Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization. *Journal of the Royal Society, Interface*, 11(97):20140352, 2014.

[42] L. Caffarelli and L. Silvestre. An extension problem related to the fractional Laplacian. *Comm. Partial Differential Equations*, 32(7-9):1245–1260, 2007.

[43] L.A. Caffarelli, J.-M. Roquejoffre, and Y. Sire. Variational problems for free boundaries for the fractional Laplacian. *J. Eur. Math. Soc. (JEMS)*, 12(5):1151–1179, 2010.

[44] L.A. Caffarelli, S. Salsa, and L. Silvestre. Regularity estimates for the solution and the free boundary of the obstacle problem for the fractional Laplacian. *Invent. Math.*, 171(2):425–461, 2008.

[45] L. Calatroni, C. Cao, J. C. De Los Reyes, C.-B. Schönlieb, and T. Valkonen. *8. Bilevel approaches for learning of variational imaging models*, pages 252 – 290. De Gruyter, Berlin, Boston, 2016.

[46] A. Carbotti, S. Dipierro, and E. Valdinoci. Local density of solutions of time and space fractional equations. *arXiv preprint arXiv:1810.08448*, 2018.

[47] B. Chang, L. Meng, E. Haber, F. Tung, and D. Begert. Multi-level residual networks from dynamical systems view. *arXiv preprint arXiv:1710.10348*, 2017.

[48] H. Chen, Q. Dou, L. Yu, J. Qin, and P-A. Heng. Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images. *NeuroImage*, 170:446–455, 2018.

[49] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He. Short-term load forecasting with deep residual networks. *IEEE Transactions on Smart Grid*, 10(4):3943–3952, July 2019.

[50] J. Chung and M.I. Español. Learning regularization parameters for general-form Tikhonov. *Inverse Problems*, 33(7):074004, 21, 2017.

[51] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333 – 338, 2012. Selected Papers from IJCNN 2011.

[52] F. H. Clarke. Generalized gradients and applications. *Trans. Amer. Math. Soc.*, 205:247–262, 1975.

[53] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93 of *Applied Mathematical Sciences*. Springer, New York, third edition, 2013.

[54] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. *Efficient Methods for Deep Neural Networks (EMDNN)*, 07 2016.

[55] Z. Di, S. Leyffer, and S. M. Wild. Optimization-based approach for joint x-ray fluorescence and transmission tomographic inversion. *SIAM Journal on Imaging Sciences*, 9(1), 1 2016.

[56] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bull. Sci. Math.*, 136(5):521–573, 2012.

[57] S. Dipierro, X. Ros-Oton, and E. Valdinoci. Nonlocal problems with Neumann boundary conditions. *Rev. Mat. Iberoam.*, 33(2):377–416, 2017.

[58] S. Dipierro, O. Savin, and E. Valdinoci. Local approximation of arbitrary functions by solutions of nonlocal equations. *The Journal of Geometric Analysis*, pages 1–28, 2016.

[59] Q. Du, M. Gunzburger, R.B. Lehoucq, and K. Zhou. A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. *Math. Models Methods Appl. Sci.*, 23(3):493–540, 2013.

[60] D. Dua and C. Graff. UCI machine learning repository, 2017.

[61] W. E. Machine learning: Mathematical theory and scientific applications. *Notices of the American Mathematical Society*, 66(11):18131820, Dec 2019.

[62] E. Esme and B. Karlik. Fuzzy c-means based support vector machines classifier for perfume recognition. *Applied Soft Computing*, 46:452 – 458, 2016.

[63] M. Felsinger, M. Kassmann, and P. Voigt. The Dirichlet problem for nonlocal operators. *Math. Z.*, 279(3-4):779–809, 2015.

[64] T. Ghosh, Y-H. Lin, and J. Xiao. The Calderón problem for variable coefficients nonlocal elliptic operators. *Comm. Partial Differential Equations*, 42(12):1923–1961, 2017.

[65] T. Ghosh, M. Salo, and G. Uhlmann. The Calderón problem for the fractional Schrödinger equation. *arXiv preprint arXiv:1609.09248*, 2016.

[66] D. A. Girard. Optimal regularized reconstruction in computerized tomography. *SIAM Journal on Scientific and Statistical Computing*, 8(6):934–950, 1987.

[67] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[68] S. Gnther, L. Ruthotto, J. Schroder, E. Cyr, and N. Gauger. Layer-parallel training of deep residual neural networks. *SIAM Journal on Mathematics of Data Science*, 2:1–23, 01 2020.

[69] G. B. Goh, N. O. Hodas, and A. Vishnu. Deep learning for computational chemistry. *Journal of Computational Chemistry*, 38(16):1291–1307, 2017.

[70] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500*, 2019.

[71] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning.* MIT press, 2016.

[72] G. Grubb. Local and nonlocal boundary conditions for μ-transmission and fractional elliptic pseudodifferential operators. *Anal. PDE*, 7(7):1649–1682, 2014.

[73] G. Grubb. Fractional Laplacians on domains, a development of Hörmander's theory of μ-transmission pseudodifferential operators. *Adv. Math.*, 268:478–528, 2015.

[74] G. Grubb. Regularity of spectral fractional Dirichlet and Neumann problems. *Math. Nachr.*, 289(7):831–844, 2016.

[75] M. Gulian, M. Raissi, P. Perdikaris, and G.E. Karniadakis. Machine learning of space-fractional differential equations. *SIAM Journal on Scientific Computing*, 41(4):A2485–A2509, 2019.

[76] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

[77] K. Hamalainen, A. Kallonen, V. Kolehmainen, M. Lassas, K. Niinimaki, and S. Siltanen. Sparse tomography. *SIAM Journal on Scientific Computing*, 35(3):B644–B665, 2013.

[78] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated mri data. *Magnetic Resonance in Medicine*, 79(6):3055–3071, 6 2018.

[79] P. C. Hansen. Regularization, gsvd and truncated gsvd (generalized singular value decomposition). *BIT Numerical Mathematics*, 29(3):491–504, 9 1988.

[80] P. C. Hansen. Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(1):1–35, Mar 1994.

[81] P. C. Hansen and D.P. OLeary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, 1993.

[82] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[83] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[84] C. F. Higham and D. J. Higham. Deep learning: an introduction for applied mathematicians. *SIAM Rev.*, 61(4):860–891, 2019.

[85] M. Hintermüller and C.N. Rautenberg. Optimal selection of the regularization function in a weighted total variation model. Part I: Modelling and theory. *J. Math. Imaging Vision*, 59(3):498–514, 2017.

[86] M. Hintermüller, C.N. Rautenberg, T. Wu, and A. Langer. Optimal selection of the regularization function in a weighted total variation model. Part II: Algorithm, its analysis and numerical tests. *J. Math. Imaging Vision*, 59(3):515–533, 2017.

[87] J. Hsieh, B. Nett, Z. Yu, K. Sauer, J-B. Thibault, and C. A. Bouman. Recent advances in CT image reconstruction. *Current Radiology Reports*, 1(1):39–51, 2013.

[88] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[89] M. Imaizumi and K. Fukumizu. Deep neural networks learn non-smooth functions effectively. *arXiv preprint arXiv:1802.04474*, 2018.

[90] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML15, page 448456. JMLR.org, 2015.

[91] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, Sep. 2017.

[92] A. C. Kak, M. Slaney, and G. Wang. Principles of computerized tomographic imaging. *Medical Physics*, 29(1):107–107, 2002.

[93] C. T. Kelley. *Iterative methods for optimization.* Frontiers in applied mathematics. SIAM, Philadelphia, 1999.

[94] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo. *Theory and applications of fractional differential equations*, volume 204 of *North-Holland Mathematics Studies.* Elsevier Science B.V., Amsterdam, 2006.

[95] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and their Applications.* Academic Press, New York, 1980.

[96] N.V. Krylov. On the paper: ”all functions are locally s-harmonic up to a small error” by Dipierro, Savin, and Valdinoci. *arXiv preprint arXiv:1810.07648*, 2018.

[97] R.-Y. Lai and Y.-H. Lin. Global uniqueness for the fractional semilinear Schrödinger equation. *Proc. Amer. Math. Soc.*, 147(3):1189–1199, 2019.

[98] P.A. Larkin and M. Whalen. Direct, near field acoustic testing. Technical report, SAE technical paper, 1999.

[99] M. Lassas and S. Siltanen. Can one use total variation prior for edge-preserving Bayesian inversion? *Inverse Problems*, 20(5):1537, 2004.

[100] D. Lee, J. Yoo, S. Tak, and J. C. Ye. Deep residual learning for accelerated mri using magnitude and phase networks. *IEEE Transactions on Biomedical Engineering*, 65(9):1985–1995, 2018.

[101] J.-L. Lions. *Optimal control of systems governed by partial differential equations.* Translated from the French by S. K. Mitter. Die Grundlehren der mathematischen Wissenschaften, Band 170. Springer-Verlag, New York-Berlin, 1971.

[102] W. Liu, X. Ma, Y. Zhou, D. Tao, and J. Cheng. p -laplacian regularization for scene recognition. *IEEE Transactions on Cybernetics*, 49(8):2927–2940, Aug 2019.

[103] C. Louis-Rose and M. Warma. Approximate controllability from the exterior of space-time fractional wave equations. *Appl Math Optim*, pages 1–44, 2018.

[104] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations, 2017.

[105] A.S. Lübbe, C. Bergemann, H. Riess, F. Schriever, P. Reichardt, K. Possinger, M. Matthias, B. Dörken, F. Herrmann, R. Gürtler, et al. Clinical experiences with magnetic drug targeting: a phase i study with 4'-epidoxorubicin in 14 patients with advanced solid tumors. *Cancer research*, 56(20):4686–4693, 1996.

[106] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, Jan 2018.

[107] J. Magiera, D. Ray, J.S. Hesthaven, and C. Rohde. Constraint-aware neural networks for riemann problems. *Journal of Computational Physics*, 409, 2020.

[108] S. Mallat and I. Waldspurger. Deep learning by scattering. *arXiv preprint arXiv:1306.5532*, 2013.

[109] M. T. McCann, K. H. Jin, and M. Unser. Convolutional Neural Networks for Inverse Problems in Imaging: A Review. *IEEE Signal Processing Magazine*, 34:85–95, November 2017.

[110] R. Metzler and J. Klafter. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics reports*, 339(1):1–77, 2000.

[111] S. G. Nash. A survey of truncated-Newton methods. *J. of Comp. and App. Math.*, 124:45–59, 2000.

[112] E. Niedermeyer and F.H.L. da Silva. *Electroencephalography: basic principles, clinical applications, and related fields.* Lippincott Williams & Wilkins, 2005.

[113] K. Niinimaki, M. Lassas, K. Hamalainen, A. Kallonen, V. Kolehmainen, E. Niemi, and S. Siltanen. Multiresolution parameter choice method for total variation regularized tomography. *SIAM Journal on Imaging Sciences*, 9(3):938–974, 2016.

[114] J. Nocedal and S. Wright. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, New York, December 2006.

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

[116] I. Podlubny. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications.* Mathematics in Science and Engineering. Academic Press, London, 1999.

[117] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–16, 2016.

[118] J. Radon. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging*, 5(4):170–176, December 1986.

[119] 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.

[120] X. Ros-Oton and J. Serra. The extremal solution for the fractional Laplacian. *Calc. Var. Partial Differential Equations*, 50(3-4):723–750, 2014.

[121] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

[122] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D Nonlinear Phenomena*, 60:259–268, November 1992.

[123] A. Rüland and M. Salo. The fractional Calderón problem: low regularity and stability. *arXiv preprint arXiv:1708.06294*, 2017.

[124] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 2019.

[125] S. G. Samko, A. A. Kilbas, and O. I. Marichev. *Fractional integrals and derivatives.* Gordon and Breach Science Publishers, Yverdon, 1993.

[126] R. Scherer. *Computer vision methods for fast image classification and retrieval.* Springer, 2020.

[127] CB. Schöenlieb, M. Benning, M. Ehrhardt, B. Owren, and E. Celledoni. Research data supporting "deep learning as optimal control problems". Dataset, 2019.

[128] R. Servadei and E. Valdinoci. On the spectrum of two different fractional operators. *Proc. Roy. Soc. Edinburgh Sect. A*, 144(4):831–855, 2014.

[129] H. Shan, A. Padole, F. Homayounieh, U. Kruger, R. D. Khera, C. Nitiwarangkul, M. K. Kalra, and G. Wang. Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose ct image reconstruction. *Nature Machine Intelligence*, 1(6):269, 2019.

[130] J. Shen and T. F. Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[131] J. Sjberg and L. Ljung. Overtraining, regularization, and searching for minimum in neural networks. *IFAC Proceedings Volumes*, 25(14):73 – 78, 1992. 4th IFAC Symposium on Adaptive Systems in Control and Signal Processing 1992, Grenoble, France, 1-3 July.

[132] J. Sprekels and E. Valdinoci. A new type of identification problems: optimizing the fractional order in a nonlocal evolution equation. *SIAM J. Control Optim.*, 55(1):70–93, 2017.

[133] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc., 2015.

[134] J.-L. Starck, E.J. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684, 2002.

[135] P. R. Stinga and J. L. Torrea. Extension problem and Harnack's inequality for some fractional operators. *Comm. Part. Diff. Eqs.*, 35(11):2092–2122, 2010.

[136] M. Stynes. Too much regularity may force too much uniqueness. *Fract. Calc. Appl. Anal.*, 19(6):1554–1562, 2016.

[137] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2798, July 2017.

[138] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems.* V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York-Toronto, Ont.-London, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[139] F. Tröltzsch. *Optimal control of partial differential equations*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2010. Theory, methods and applications, Translated from the 2005 German original by Jürgen Sprekels.

[140] M. Unsworth. New developments in conventional hydrocarbon exploration with electromagnetic methods. *CSEG Recorder*, 30(4):34–38, 2005.

[141] E. Valdinoci. From the long jump random walk to the fractional Laplacian. *Bol. Soc. Esp. Mat. Apl. SeMA*, (49):33–44, 2009.

[142] A. Veit, M. J. Wilber, and S. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pages 550–558, 2016.

[143] M.I. Visik and G.I. Èskin. Convolution equations in a bounded region. *Uspehi Mat. Nauk*, 20(3 (123)):89–152, 1965.

[144] C. R. Vogel. Non-convergence of the *L*-curve regularization parameter selection method. *Inverse Problems*, 12(4):535–547, 1996.

[145] M. Warma. A fractional Dirichlet-to-Neumann operator on bounded Lipschitz domains. *Commun. Pure Appl. Anal.*, 14(5):2043–2067, 2015.

[146] M. Warma. The fractional relative capacity and the fractional Laplacian with Neumann and Robin boundary conditions on open sets. *Potential Anal.*, 42(2):499–547, 2015.

[147] M. Warma. Approximate controllabilty from the exterior of space-time fractional diffusive equations. *SIAM Journal on Control and Optimization (SICON)*, 2019, to appear.

[148] M. Warma and S. Zamorano. Analysis of the controllability from the exterior of strong damping nonlocal wave equations. *arXiv preprint arXiv:1810.08060*, 2018.

[149] C. J. Weiss, B.G. van Bloemen Waanders, and H. Antil. Fractional operators applied to geophysical electromagnetics. *Geophysical Journal International*, 220(2):1242–1259, 2020.

[150] A. Wigderson. *Mathematics and Computation.* Princeton University Press, Princeton, 2019.

[151] R.L. Williams, I. Karacan, and C.J. Hursch. *Electroencephalography (EEG) of human sleep: clinical applications.* John Wiley & Sons, 1974.

[152] S. Wu, S. Zhong, and Y. Liu. Deep residual learning for image steganalysis. *Multimedia tools and applications*, 77(9):10437–10453, 2018.

[153] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep admm-net for compressive sensing mri. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 10–18, USA, 2016. Curran Associates Inc.

[154] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, July 2017.

[155] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei. Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4274–4288, 2018.

[156] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018.

BIOGRAPHY

Ratna Khatri

Ratna Khatri received her Bachelor of Science in Mathematics from George Mason University in 2016. She received her Ph.D. in Mathematics from George Mason University in 2020. In 2017, she was a Givens Associate at Argonne National Laboratory. As a student at George Mason University, she was an active member of the Association for Women in Mathematics and Society for Industrial and Applied Mathematics student chapters. After her Ph.D., she joined the U.S. Naval Research Laboratory as a Research Scientist.