Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography

IOP Publishing Inverse Problems
Inverse Problems **36** (2020) 064001 (22pp) https://doi.org/10.1088/1361-6420/ab80d7

Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography

Harbir Antil[1,3], **Zichao Wendy Di**[2] and **Ratna Khatri**[1]

[1] Department of Mathematical Sciences and The Center for Mathematics and Artificial Intelligence (CMAI), George Mason University, Fairfax, VA 22030, United States of America
[2] Mathematics and Computer Science Division, Argonne National Laboratory, IL 60439, United States of America

E-mail: hantil@gmu.edu, wendydi@mcs.anl.gov and rkhatri3@gmu.edu

Received 5 November 2019, revised 26 January 2020
Accepted for publication 18 March 2020
Published 29 April 2020

Abstract

In this work we consider a generalized bilevel optimization framework for solving inverse problems. We introduce fractional Laplacian as a regularizer to improve the reconstruction quality, and compare it with the total variation regularization. We emphasize that the key advantage of using fractional Laplacian as a regularizer is that it leads to a linear operator, as opposed to the total variation regularization which results in a nonlinear degenerate operator. 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 general regularized inverse problem. We illustrate how to incorporate various regularizer choices into our proposed network. As an example, we consider tomographic reconstruction as a model problem and show an improvement in 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 observe that the fractional Laplacian regularization outperforms total variation regularization. This is specially encouraging, and important, in the case of limited and noisy data.

Keywords: bilevel optimization neural network, fractional Laplacian regularization, deep residual learning, imaging science, tomographic reconstruction, inverse problems

(Some figures may appear in colour only in the online journal)

[3] Author to whom any correspondence should be addressed.

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 [1–5]. 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 [6, 7] is of edge-preserving nature. However, its non-differentiability makes its usage numerically challenging. Another choice is the Tikhonov regularization [8], 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 (RNN) have shown remarkable potential in image classification and reconstruction where, often, the goal is to learn the whole regularizer [9–11]. These approaches, however, may not be robust in general [12, 13]. 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 into the network, for instance. In addition, they often lack a mathematical justification [14–18]. The main contributions of this paper are two-folds:

(a) Extend the fractional Laplacian introduced in [19] 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 [20]), see also [21, 22].

Our approach is closely related to the methodology introduced in [9]. 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 [23] for a discussion on bilevel optimization. We emphasize that the main novelty in our paper is the use of fractional Laplacian [19, 24, 25] 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 [26] 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 [27]. 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 [19, 28], geophysics [29], diffusion maps [30], biology [31], novel exterior optimal control [32, 33], 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 [34, 35]. 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 [36], and the references therein.

The numerical examples presented in this paper are strongly motivated by tomographic reconstruction, see sub section 2.3. Further realistic application of interest to us is the MRI reconstruction, considered in [9]. 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 future work.

The rest of the paper is organized as follows. In section 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 3 is devoted to our proposed algorithmic framework, i.e., the *bilevel optimization neural network* (BONNet) to learn the optimal regularization strength, as well as the order of the fractional Laplacian. In section 4, we provide further numerical experiments illustrating the application of BONNet to the tomographic reconstruction problem.

2. Regularization in inverse problems

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

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

where $f : \Omega \mapsto \mathbb{R}$ is a given function and $\Omega \subset \mathbb{R}^n$ with $n \geqslant 1$ 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 (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 $n \geqslant 1$ 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

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

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) [6], convolution operator (image deblurring problem) [37, 38], and the Fourier or wavelet transforms [39]. Therefore, in (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) may be ill-posed [40]. 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.

2.1. Total variation regularization

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

$$\mathcal{R}(u,\mu) = \lambda\,\mathrm{TV}(u), \tag{3}$$

where $\mu = \lambda$ is a scalar. Here, $\mathrm{TV}(u)$ denotes the total variation semi-norm on Ω and $X = \mathrm{BV}(\Omega)\cap L^2(\Omega)$, where $\mathrm{BV}(\Omega)$ denotes the set of functions of bounded variations [41]. Formally speaking, $\mathrm{TV}(u) := \int_\Omega |\nabla u|$ and as a result the corresponding Euler–Lagrange equation for (2) is: find $u \in X_{\mathrm{ad}} \subset X$ such that

$$\left\langle -\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + K^*(Ku-f), \hat{u}-u\right\rangle_{X',X} \geqslant 0, \quad \forall\, \hat{u}\in X_{\mathrm{ad}} \tag{4}$$

i.e., a nonlinear and possibly degenerate (due to $1/|\nabla u|$) variational equation which is challenging to solve. We remark that X' is the dual of X and K^* is the adjoint of K. Designing solvers for 4 is still an active area of research [42]. 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 [6].

2.2. Fractional Laplacian regularization

The fractional Laplacian as a regularization for (2) is given by,

$$\mathcal{R}(u,\mu) = \frac{1}{2}\|\sqrt{\lambda}(-\Delta)^{\frac{s}{2}}u\|^2_{L^2(\Omega)}, \tag{5}$$

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 [19, 25]. 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 (5) is: find $u \in X_{\mathrm{ad}}$

$$\langle \lambda\,(-\Delta)^s u + K^*(Ku-f), \hat{u}-u\rangle \geqslant 0, \quad \forall\, \hat{u}\in X_{\mathrm{ad}} \tag{6}$$

i.e., a variational equation with a linear operator. Such a problem has a unique solution in the fractional order Sobolev space $X = H^s(\Omega)$ [43]. This regularization has been applied successfully in image denoising [19] (with $K = I$, but with $u \in X$, instead of X_{ad}, as a result (6) becomes an equality).

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

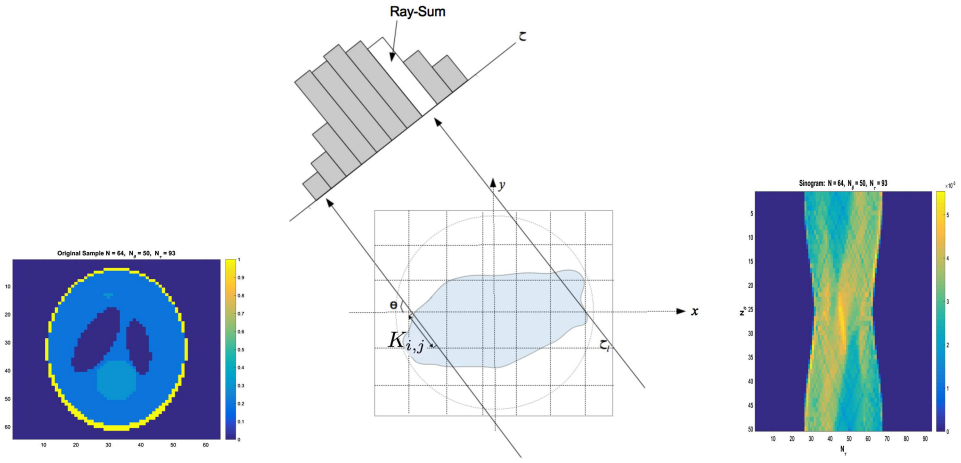

Figure 1. Geometric sketch of x-ray tomography (middle) which maps the sample (left) from the (x, y) space to the sinogram (*right*) on the (τ, θ) space.

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 [44] (see figure 1). However, the limited data, due to the discrete nature of the physical experiment and dosage limits, 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 [45], for which K is defined as,

$$Ku(\tau, \theta) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, y)\delta(\tau - x\cos\theta - y\sin\theta)\,\mathrm{d}x\,\mathrm{d}y, \tag{7}$$

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 cannot 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.

2.4. 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 1. To get the noisy data, we add 0.1% Gaussian noise to f. More details on tomographic reconstruction is provided in section 4. Next we show the reconstructions based on the two

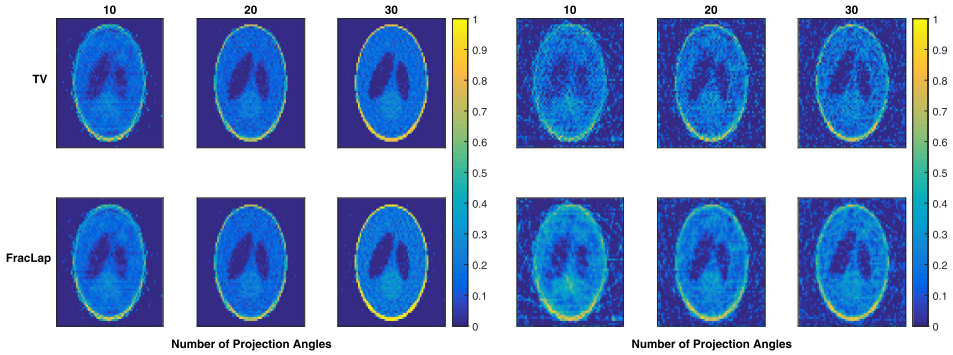

Figure 2. Tomographic reconstructions based on the total variation regularization (*row 1*) and fractional Laplacian (with $s = 0.4$, *row 2*) for data without noise (*left*) and with 0.1% noise (*right*). The fractional Laplacian outperforms the total variation regularization in recovering finer features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

regularizers, namely the fractional Laplacian (5) and the total variation (3), in figure 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 [46]. 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 (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' only complicates the parameter choice further.

For the reconstructions in figure 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 [19]), and solve the minimization problem (2) using an inexact truncated-Newton method for bound-constrained problems [47]. The optimal value of λ is then chosen using a combination of L-curve criterion [48] 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 [49].

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

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 1. We first state a generic bilevel optimization problem,

$$\begin{aligned} &\min_{\mu\in\mathcal{M}_{\text{ad}}} \quad \phi(\mu) \\ &\min_{u\in X_{\text{ad}}} J(u,\mu) := \frac{1}{2}\|Ku - f\|^2_{L^2(\Omega)} + \mathcal{R}(u,\mu), \end{aligned} \tag{8}$$

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

In section 3.1, motivated by [9], 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 paper is to use fractional Laplacian as the regularizer. Notice that the lower level problem (2) in (8) can be solved using the existing techniques.

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 [50]. Deep residual learning has enabled remarkable progress in imaging science [21, 50, 51], biomedical applications [9, 52, 53], satellite imagery, remote sensing [54–56], 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 (8) for which we seek our modeling inspiration from [9], and define $\phi(\mu)$ as the average mean squared error over m distinct samples, i.e.,

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

where $u(\mu)$ solves the lower level problem in (8), 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 [21, 22]. 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 (8) which depends on μ. Notice that the inner problem in (8) is same as (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,

$$\mathcal{R}(u,\mu) := \mathcal{R}(\sigma(T)) = \frac{1}{2}\|\sigma(T(\mu, u(\mu)))\|^2_{L^2(\Omega)}. \tag{9}$$

Then, for m distinct samples, we can write our inner minimization problem (2) with a generalized regularizer as an average over m samples, and $\mu \in \mathcal{M}_{\text{ad}}$,

$$\min_{u\in X_{\text{ad}}} \quad J(u,\mu) := \frac{1}{2m}\sum_{i=1}^{m}\left[\|Ku^{(i)} - f^{(i)}\|^2_{L^2(\Omega)} + \|\sigma\|^2_{L^2(\Omega)}\right]. \tag{10}$$

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 (10) w.r.t. u in its variational form is; for each sample, $i = 1, \ldots, m$,

$$\begin{aligned} DJ(u^{(i)},\mu)[h] = \frac{1}{m}\Big[&(K^*(Ku^{(i)} - f^{(i)}), h)_{L^2(\Omega)} \\ &+ \left(\left(\partial_{u^{(i)}} T\right)^*(\partial_T \sigma)\sigma, h\right)_{L^2(\Omega)}\Big]. \end{aligned} \tag{11}$$

- **Solver: projected gradient descent method** The choices of X_{ad} and $\mathcal{M}_{\text{ad}}$ are problem dependent, for example, for tomographic reconstruction model, we let $X_{\text{ad}} := \{u \in X | u \geqslant 0\}$. Moreover, we set $\mathcal{M}_{\text{ad}} := \Lambda_{\text{ad}}$ for total variation and $\mathcal{M}_{\text{ad}} := \Lambda_{\text{ad}} \times S_{\text{ad}}$ where $\Lambda_{\text{ad}} := \{\lambda \in \mathbb{R} \,|\, \lambda \geqslant \epsilon_1 > 0\}$ and $S_{\text{ad}} := \{s \in \mathbb{R} \,|\, 0 < \epsilon_2 \leqslant s \leqslant 1 - \epsilon_2\}$ for the fractional Laplacian. See section 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* [57] to solve our inner and outer minimization problems in (8). Then, the projected gradient descent scheme for solving (10), 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) $j = 1, \ldots, n$, is given by

$$u^{(i)}_j = P_{X_{\text{ad}}}\left(u^{(i)}_{j-1} - \alpha \nabla_{u^{(i)}_{j-1}} J(u^{(i)}_{j-1}, \mu)\right). \tag{12}$$

where $P_{X_{\text{ad}}}(\cdot)$ denotes the projection on the admissible set X_{ad}, see section 4.1.2 for more details on the tomographic reconstruction application. Note that, (12) is also known as the *forward propagation*. We are using ∇ to denote the gradient and D to denote the directional derivative (cf (11)). Now substitute the gradient from (11) in (12) to arrive at,

$$u^{(i)}_j = P_{X_{\text{ad}}}\left(u^{(i)}_{j-1} - \frac{\alpha}{m}\left[K^*(Ku^{(i)}_{j-1} - f^{(i)}) + (\partial_{u^{(i)}_{j-1}} T)^*(\partial_T \sigma)\sigma\right]\right). \tag{13}$$

To compute the learning rate α, we use line search for projected gradient descent as described in [57, p 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 $u^{(i)}_{\text{true}}$ and $f^{(i)}$ be the known true solution and its corresponding experimental data for the ith sample, with $i = 1, \ldots, m$. Then, we formulate our bilevel supervised learning problem as; for $j = 1, \ldots, n$,

$$\begin{aligned} \min_{\mu \in \mathcal{M}_{\text{ad}}} \phi(\mu) &= \frac{1}{2m} \sum_{i=1}^{m} \|u^{(i)}_n(\mu) - u^{(i)}_{\text{true}}\|^2_{L^2(\Omega)} \\ \text{s.t.} \quad u^{(i)}_j &= P_{X_{\text{ad}}} \left(u^{(i)}_{j-1} - \frac{\alpha}{m} [K^*(K u^{(i)}_{j-1} - f^{(i)}) + (\partial_{u^{(i)}_{j-1}} T)^*(\partial_T \sigma)\sigma] \right). \end{aligned} \tag{14}$$

Remark 3.1 (Relation to existing neural networks)**.** Notice the resemblance between the inner level problem in (14) and a residual neural network [50, 58], see also for other related works [59–61]. Indeed, after rewriting we obtain that

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

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

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

$$\mu_{l+1} = P_{\mathcal{M}_{\text{ad}}} \left(\mu_l - \beta \nabla_{\mu_l} \phi(\mu_l) \right), \qquad l = 0, \ldots, q-1, \tag{15}$$

where $P_{\mathcal{M}_{\text{ad}}}(\cdot)$ 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 $u^{(i)}_n$ is sufficiently smooth with respect to μ. After applying the chain rule, we obtain that

$$\nabla_{\mu_l} \phi(\mu_l) = \frac{1}{m} \sum_{i=1}^{m} \int_\Omega (u^{(i)}_n - u^{(i)}_{\text{true}}) \frac{\mathrm{d}u^{(i)}_n}{\mathrm{d}\mu}\bigg|_{\mu=\mu_l} \mathrm{d}\Omega. \tag{16}$$

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 (14). We evaluate $\frac{\mathrm{d}u^{(i)}_n}{\mathrm{d}\mu}\Big|_{\mu=\mu_l}$ in (16) 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 $j = 1, \ldots, n$

$$\frac{\mathrm{d}u_j}{\mathrm{d}\mu}\bigg|_{\mu=\mu_l} = \frac{\partial u_j}{\partial u_{j-1}} \cdot \frac{\mathrm{d}u_{j-1}}{\mathrm{d}\mu}\bigg|_{\mu=\mu_l} + \frac{\partial u_j}{\partial \mu} \cdot \frac{\mathrm{d}\mu}{\mathrm{d}\mu}\bigg|_{\mu=\mu_l}, \tag{17}$$

where,

$$\frac{\partial u_j}{\partial u_{j-1}} = I - \frac{\alpha}{m}\left[K^*K + \frac{\partial}{\partial u_{j-1}}\left(\frac{\partial T}{\partial u_{j-1}}\right)^*\left(\frac{\partial \sigma}{\partial T}\right)^*\sigma + \left(\frac{\partial T}{\partial u_{j-1}}\right)^*\frac{\partial}{\partial u_{j-1}}\left(\frac{\partial \sigma}{\partial T}\right)^*\sigma + \left(\frac{\partial T}{\partial u_{j-1}}\right)^*\left(\frac{\partial \sigma}{\partial T}\right)^*\left(\frac{\partial \sigma}{\partial T}\cdot\frac{\partial T}{\partial u_{j-1}}\right)\right], \tag{18}$$

and,

$$\frac{\partial u_j}{\partial \mu} = -\frac{\alpha}{m}\left[\left(\frac{\partial}{\partial \mu}\left(\frac{\partial T}{\partial u_{j-1}}\right)^*\right)\left(\frac{\partial \sigma}{\partial T}\right)^*\sigma + \left(\frac{\partial T}{\partial u_{j-1}}\right)^*\left(\frac{\partial}{\partial \mu}\left(\frac{\partial \sigma}{\partial T}\right)^*\right)\cdot\sigma + \left(\frac{\partial T}{\partial u_{j-1}}\right)^*\left(\frac{\partial \sigma}{\partial T}\right)^*\cdot\frac{\partial \sigma}{\partial T}\frac{\partial T}{\partial \mu}\right]. \tag{19}$$

Substituting (18) and (19) in (17) 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 (14) to learn the regularization parameters, and during the *testing* phase we only solve the inner problem in (14) 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).

3.1.1. General framework of BONNet. We summarize the training and testing phases of our deep BONNet architecture as follows:

- ***TrainingPhase*** **(algorithm 1).** In this phase, we pass in m training samples $\left\{u^{(i)}_{\text{true}}, f^{(i)}\right\}_{i=1}^{m}$ 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.
- ***TestingPhase*** **(algorithm 2).** In this phase, we use the μ^* learned from the *training* phase and testing data $\left\{f^{(i)}_{\text{test}}\right\}_{i=1}^{m_{\text{test}}}$ 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.

Remark 3.2 (Fixed vs variable depth of BONNet)**.** We remark that instead of specifying the number of layers when solving (15) or (13), 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 [57, p 91] for both μ and u. Also note that for (13), 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, $n << n_{\text{test}}$ prevents the network from *overfitting* of parameters to the training data, and helps the model generalize to unseen data [62]. 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.

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

Algorithm 1. Training Phase of BONNet.

Input: $\left\{u^{(i)}_{\text{true}}, f^{(i)}\right\}_{i=1}^{m}$, m training samples
Output: μ^*

1: Initialize u_0, $\frac{\mathrm{d}u_0}{\mathrm{d}\mu}$ and μ_0
2: **for** for $l = 0$ to $q-1$ **do**
3: **for** for $j = 1$ to n **do**
4: Compute $u^{(i)}$ and $\frac{\mathrm{d}u^{(i)}_n}{\mathrm{d}\mu}$ for all $i = 1, \ldots, m$:

$$u^{(i)}_j = P_{X_{\text{ad}}}\left(u^{(i)}_{j-1} - \frac{\alpha}{m}\left[K^*(Ku^{(i)}_{j-1} - f^{(i)}) + (\partial_{u^{(i)}_{j-1}} T)^*(\partial_T \sigma)\sigma\right]\right).$$

{Compute α using line search as discussed in section 3.1}

$$\left.\frac{\mathrm{d}u^{(i)}_j}{\mathrm{d}\mu}\right|_{\mu=\mu_l} = \frac{\partial u^{(i)}_j}{\partial u^{(i)}_{j-1}} \cdot \left.\frac{\mathrm{d}u^{(i)}_{j-1}}{\mathrm{d}\mu}\right|_{\mu=\mu_l} + \frac{\partial u^{(i)}_j}{\partial \mu} \cdot \left.\frac{\mathrm{d}\mu}{\mathrm{d}\mu}\right|_{\mu=\mu_l}$$

{See (18), (19) for explicit expressions}
5: **end for**
6: Compute the gradient of $\phi(\mu)$:

$$\nabla_{\mu_l}\phi(\mu_l) = \frac{1}{m}\sum_{i=1}^{m}\int_\Omega (u^{(i)}_n - u^{(i)}_{\text{true}}) \left.\frac{\mathrm{d}u^{(i)}_n}{\mathrm{d}\mu}\right|_{\mu=\mu_l} \mathrm{d}\Omega,$$

7: Update μ:

$$\mu_{l+1} = P_{\mathcal{M}_{\text{ad}}}\left(\mu_l - \beta\nabla_{\mu_l}\phi(\mu_l)\right).$$

{Compute β using line search as discussed in section 3.1}
8: **end for**

framework (9). In section 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 (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 (9), 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, (14), (18), and (19) become, for $j = 1, \ldots, n$,

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

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

and

$$\frac{\partial u_j}{\partial \lambda} = -\frac{\alpha}{m}(-\Delta)^s u_{j-1}, \quad \text{and} \quad \frac{\partial u_j}{\partial s} = -\frac{\alpha\lambda}{m}\frac{\partial}{\partial s}((-\Delta)^s u_{j-1}) \tag{20}$$

which together give us the sensitivity of u w.r.t. μ in (17). Notice that the second equation in (20) requires the sensitivity of fractional Laplacian $(-\Delta)^s$ with respect to 's'. This is

Algorithm 2. Testing Phase of BONNet.

Input: μ^*, $\left\{f_{\text{test}}^{(i)}\right\}_{i=1}^{m_{\text{test}}}$, m_{test} testing samples
Output: u
1: Initialize u_0
2: **for** for $j = 1$ to n_{test} **do**
3: Compute u for all $i = 1, \ldots, m_{\text{test}}$:

$$u_j^{(i)} = P_{X_{\text{ad}}}\left(u_{j-1}^{(i)} - \frac{\alpha}{m}\left[K^*(Ku_{j-1}^{(i)} - f_{\text{test}}^{(i)}) + (\partial_{u_{j-1}^{(i)}} T)^*(\partial_T \sigma)\sigma\right]\right).$$

{Compute α using line search as discussed in 3.1}
4: **end for**

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

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

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

$$\text{TV}_\xi(u) = \int_\Omega \sqrt{|\nabla u|^2_{\ell^2(\Omega)} + \xi^2}\, \partial\Omega. \tag{21}$$

with $0 < \xi \ll 1$. We will omit the subscript ξ from TV_ξ for brevity. Then, to define the corresponding generalized regularizer (9), let $T(\mu, u(\mu)) := 2|\Omega|^{-1}\lambda \text{TV}(u)$, and the activation function $\sigma(T) := \sqrt{T}$. Then, after some simplifications, (14), (18), and (19) become, for $j = 1, \ldots, n$,

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

$$\begin{aligned}\frac{\partial u_j}{\partial u_{j-1}} &= I - \frac{\alpha}{m}K^*K + \frac{\alpha\lambda}{2\,m}\,\text{div}\left(\frac{\partial}{\partial u_{j-1}}\left(\frac{\nabla u_{j-1}}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\right)\right)\\ &= I - \frac{\alpha}{m}K^*K + \frac{\alpha\lambda}{2\,m}\,\text{div}\left(\frac{\nabla}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\right)\\ &\quad + \frac{\alpha\lambda}{2\,m}\,\text{div}\left(\nabla u_{j-1}\frac{\partial}{\partial u_{j-1}}\left(\frac{1}{\sqrt{|\nabla u_{j-1}|^2_{\ell^2(\Omega)} + \xi^2}}\right)\right),\end{aligned} \tag{22}$$

and

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

which together give us the sensitivity of u w.r.t. μ in (17). Again, we have omitted the superscript 'i' to improve readability.

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 section 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 remark 3.2, we run the proposed algorithm until a desired tolerance (tol) is met. At the testing phase we set tol $= 1 \times 10^{-5}$ and at the training phase we set 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 (21) as $\xi = 1 \times 10^{-5}$. In our numerical examples, we have noticed that the last term in (22) 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 section 4.1 we discuss the implementation details of fractional Laplacian and the admissible sets X_{ad} and $\mathcal{M}_{\text{ad}}$. This is followed by two experiments in section 4.2.

4.1. Preliminaries

Before we discuss the actual results, we state some preliminary material. As mentioned in the paragraph following (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.

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 (20) we also recall that we need to approximate the variation of $\mathbf{A}^s$ with respect to 's'. A straightforward calculation gives

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

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 [63] and the references therein.

4.1.2. Admissible sets and projection. For tomographic reconstruction we let $X_{\text{ad}} := \{u \in X | u \geqslant 0\}$. Moreover, we set $\mathcal{M}_{\text{ad}} := \Lambda_{\text{ad}}$ for total variation and $\mathcal{M}_{\text{ad}} := \Lambda_{\text{ad}} \times S_{\text{ad}}$ where

$\Lambda_{\text{ad}} := \{\lambda \in \mathbb{R} \,|\, \lambda \geqslant \epsilon_1 > 0\}$ and $S_{\text{ad}} := \{s \in \mathbb{R} \,|\, 0 < \epsilon_2 \leqslant s \leqslant 1 - \epsilon_2\}$. We let $\epsilon_1 = \epsilon_2 = 10^{-15}$.

Furthermore, the projection in (13) onto the admissible set X_{ad} is given by, for $z \in X$,

$$P_{X_{\text{ad}}}(z) := \max\{0, z\} = \begin{cases} z & \text{if} \quad z \geqslant 0, \\ 0 & \text{if} \quad z < 0. \end{cases} \tag{23}$$

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

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

For a rigorous definition of the generalized derivative of the *max* function, see [64]. Similar projection formulas are applicable for projection onto the set $\mathcal{M}_{\text{ad}}$.

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 [57]. 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 section 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.

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 3 for two representative samples). We use a convention of choosing $N_\tau \geqslant \sqrt{2}N$ 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 [65]. 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 $m_{\text{test}} = 10$ testing samples.

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 algorithms 1 and 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 $\mu = (\lambda, s)$.

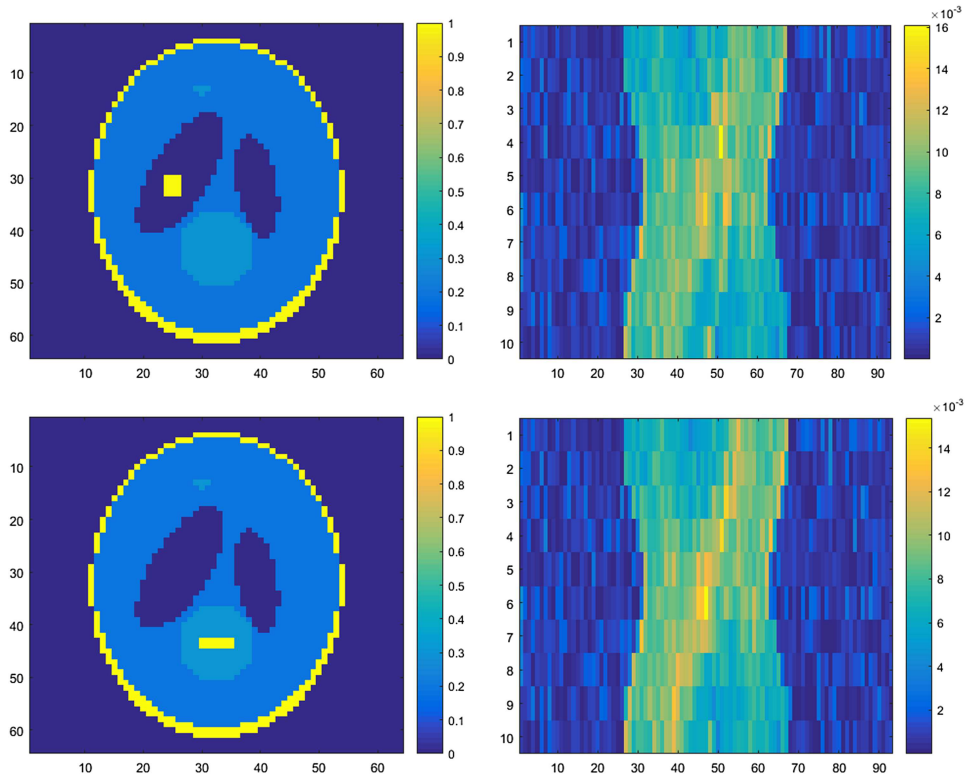

Figure 3. Representative samples of Phantom (u_{true}) used (*left*) to generate the synthetic data (noisy sinogram f) (*right*) for training (*row 1*) and testing (*row 2*).

4.2.1. Results of experiment I: learning λ, fixed $s = 0.4$. We now discuss the results of our experiments. In figure 4, we compare the reconstructions obtained from BONNet with the true solution shown in figure 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, $\{(u^{(i)}_{\text{true}}, f^{(i)}_{\text{train}})\}_{i=1}^{m=20}$ are passed to the deep BONNet algorithm 1. The λ values mentioned under each reconstruction are the corresponding optimal λ^*_{none}, λ^*_{TV}, and $\lambda^*_{\text{fracLap}}$ that we learn during the training stage. Notice that $\lambda^*_{\text{none}} = 0$ corresponds to 'no regularization'. The *right panel* corresponds to the reconstructions at the n_{test}th layer of the testing phase. Recall that $\{(\lambda^*, f^{(i)}_{\text{test}})\}_{i=1}^{m_{\text{test}}=10}$ are passed to the deep BONNet at this stage in algorithm 2.

From the reconstructions in figure 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 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

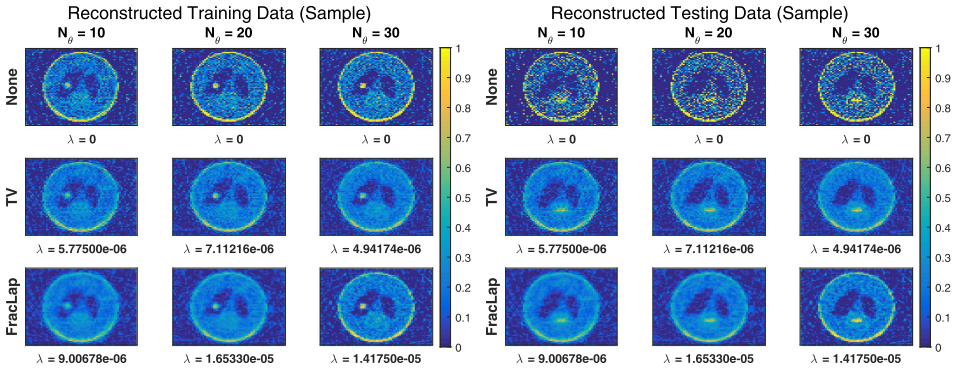

Figure 4. Comparison of reconstructions based on various regularizers (*rows*) and various number of tomographic projection angles (*columns*) for data with 0.1% Gaussian noise. The *left* and *right* panels correspond to the solution at the last layer for two of the many distinct samples used during 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.

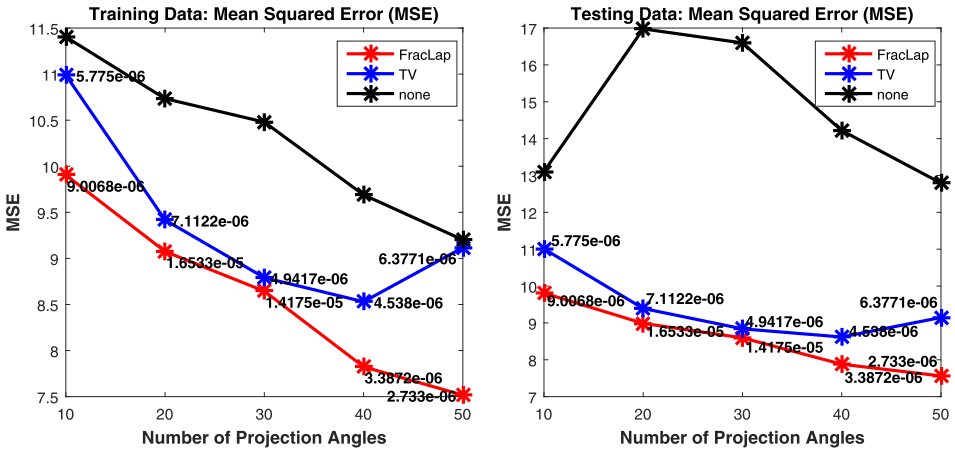

Figure 5. We compare the mean-squared errors (MSE) for the solution, averaged over 20 training (respectively, 10 testing) samples [*left*(respectively, right)], against various number of projection angles for the tomographic reconstruction problem. The solid *black*, *blue* and *red* lines corresponds 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.

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 nonlinear.

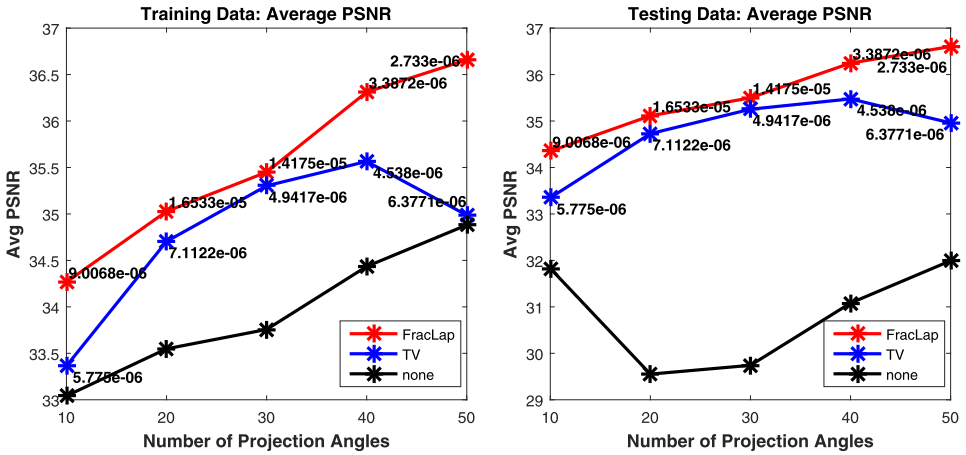

Figure 6. We compare the peak signal-to-noise ratio (PSNR) for the solution, averaged over 20 training (respectively, 10 testing) samples [*left*(respectively, right)], against various number of projection angles for the tomographic reconstruction problem. The solid *black*, *blue* and *red* lines corresponds 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.

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 5, *peak signal-to-noise ratio* (PSNR) figure 6, and *structural similarity index* (SSIM) figure 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 using the inexact truncated-Newton method for bound-constrained problem [47]. We emphasize that one may use a different solver during the testing stage once λ^* is obtained via BONNet training.

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 section 3.1.2 and use the same training and testing data as described in the previous example. In table 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 section 4.2.1. In the case of $N_\theta = 10$, we obtain $(\lambda^*_{\text{fracLap}}, s^*) = (5.04417 \times$

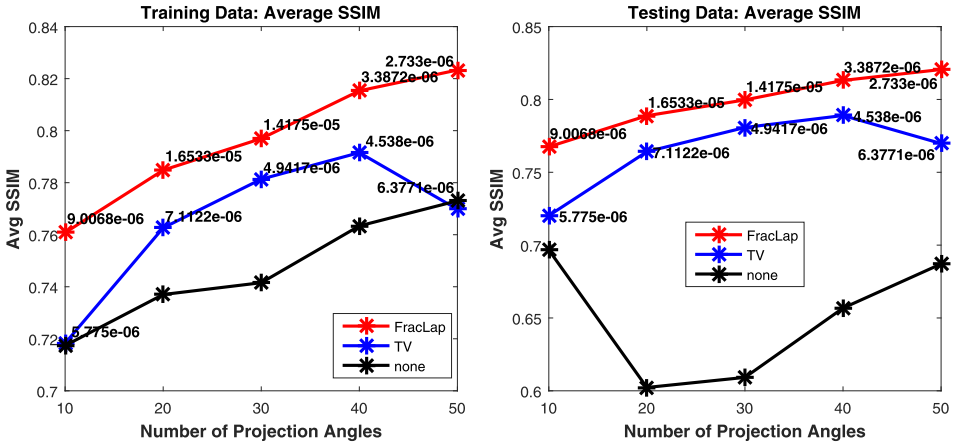

Figure 7. We compare the peak structural similarity (SSIM) for the solution, averaged over 20 training (respectively, 10 testing) samples (*left*(respectively, right)), against various number of projection angles for the tomographic reconstruction problem. The solid *black*, *blue* and *red* lines corresponds 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.

Table 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)	$(9.006\,78 \times 10^{-6}$, 0.4)	$(5.044\,17 \times 10^{-6}$, 0.5413)	$(1.653\,30 \times 10^{-5}$, 0.4)	$(8.537\,17 \times 10^{-6}$, 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

10^{-6}, 0.5413) and in the case of $N_\theta = 20$, we obtain $(\lambda^*_{\text{fracLap}}, s^*) = (8.53717 \times 10^{-6}, 0.3799)$. The reconstructions of u with $(\lambda^*_{\text{fracLap}}, s^*)$ are visually comparable to the case of fractional Lapalcian in figure 4 and therefore they have been omitted. We observe that all the error metrics returned by BONNet are either comparable, or slightly better, than the ones obtained by BONNet for a fixed 's', discussed in section 4.2.1. The advantage now is that we no longer need to tune the parameters manually.

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.

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 $(\lambda^*_{\text{fracLap}}, s^*)$ pair for the fractional Laplacian regularizer, and this indicates the flexibility of our proposed network to learn non-standard parameters.

Acknowledgments

The first and third authors are partially supported by NSF grants DMS-1818772, DMS-1913004, the Air Force Office of Scientific Research under Award No.: FA9550-19-1-0036, and the Department of Navy, Naval PostGraduate School under Award No.: N00244-20-1-0005. The third author is also partially supported by a Provost award at George Mason University under the Industrial Immersion Program. The second author is partially supported by DOE Office of Science under Contract No. DE-AC02-06CH11357.

ORCID iDs

Harbir Antil https://orcid.org/0000-0002-6641-1449
Ratna Khatri https://orcid.org/0000-0003-0931-4025

References

[1] Girard D A 1987 Optimal regularized reconstruction in computerized tomography *SIAM J. Sci. Stat. Comput.* **8** 934–50

[2] Hamalainen K, Kallonen A, Kolehmainen V, Lassas M, Niinimaki K and Siltanen S 2013 Sparse tomography *SIAM J. Sci. Comput.* **35** B644–65
[3] Hsieh J, Nett B, Yu Z, Sauer K, Thibault J-B and Bouman C A 2013 Recent advances in CT image reconstruction *Current Radiology Reports* **1** 39–51
[4] Lassas M and Siltanen S 2004 Can one use total variation prior for edge-preserving Bayesian inversion? *Inverse Problems* **20** 1537
[5] Niinimaki K, Lassas M, Hamalainen K, Kallonen A, Kolehmainen V, Niemi E and Siltanen S 2016 Multiresolution parameter choice method for total variation regularized tomography *SIAM J. Imag. Sci.* **9** 938–74
[6] Rudin L, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Phys. Nonlinear Phenom.* **60** 259–68
[7] Shen J and Chan T F 2002 Mathematical models for local nontexture inpaintings *SIAM J. Appl. Math.* **62** 1019–43
[8] Tikhonov A N and Arsenin V Y 1977 *Solutions of Ill-Posed Problems* (Scripta Series in Mathematics) ed F John (Washington, D.C: V. H. Winston & Sons)
[9] Hammernik K, Klatzer T, Kobler E, Recht M P, Sodickson D K, Pock T and Knoll F 2018 Learning a variational network for reconstruction of accelerated mri data *Magn. Reson. Med.* **79** 3055–71
[10] Yang Y, Sun J, Li H and Xu Z 2016 Deep admm-net for compressive sensing mri *Proc. of the 30th Int. Conf. on Neural Information Processing Systems, NIPS'16* (USA) (Curran Associates) pp 10–8
[11] Zhang K, Zuo W, Gu S and Zhang L 2017 Learning deep CNN denoiser prior for image restoration *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2808–17
[12] Lucas A, Iliadis M, Molina R and Katsaggelos A K 2018 Using deep neural networks for inverse problems in imaging: Beyond analytical methods *IEEE Signal Process. Mag.* **35** 20–36
[13] McCann M T, Jin K H and Unser M 2017 Convolutional Neural Networks for Inverse Problems in Imaging: A Review *IEEE Signal Process. Mag.* **34** 85–95
[14] E W 2019 Machine learning: mathematical theory and scientific applications *Not. AMS* **66** 1813–20
[15] Glorot X and Bengio Y Understanding the difficulty of training deep feedforward neural networks *Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics* (Chia Laguna Resort, Sardinia, Italy 13–15 May 2010) (Proceedings of Machine Learning Research of vol 9) ed Y W Teh and M Titterington pp 249–56
[16] Qiu J, Wu Q, Ding G, Xu Y and Feng S 2016 A survey of machine learning for big data processing *EURASIP Journal on Advances in Signal Processing* **2016** 67
[17] Ruthotto L and Haber E 2019 Deep Neural Networks Motivated by Partial Differential Equations *J. Math. Imag. Vision* **62** 352–64
[18] Wigderson A 2019 Mathematics and Computation (Princeton, NJ: Princeton University Press)
[19] Antil H and Bartels S 2017 Spectral approximation of fractional pdes in image processing and phase field modeling *Comput. Methods Appl. Math.* **17** 661–78
[20] Kak A C, Slaney M and Wang G 2002 Principles of computerized tomographic imaging *Med. Phys.* **29** 107
[21] Jin K H, McCann M T, Froustey E and Unser M 2017 Deep convolutional neural network for inverse problems in imaging *IEEE Trans. Image Process.* **26** 4509–22
[22] Shan H, Padole A, Homayounieh F, Kruger U, Khera R D, Nitiwarangkul C, Kalra M K and Wang G 2019 Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose ct image reconstruction *Nature Machine Intelligence* **1** 269
[23] Calatroni L, Cao C, De Los Reyes J C, Schönlieb C-B and Valkonen T 2016 Bilevel approaches for learning of variational imaging models *Variational Methods* (Berlin: Walter de Gruyter GmbH) pp 252–90
[24] Caffarelli L and Silvestre L 2007 An extension problem related to the fractional Laplacian *Comm. Partial Differential Equations* **32** 1245–60
[25] Stinga P R and Torrea J L 2010 Extension problem and Harnack's inequality for some fractional operators *Comm. Part. Diff. Eqs.* **35** 2092–122
[26] Chung J and Español M I 2017 Learning regularization parameters for general-form Tikhonov *Inverse Problems* **33** 074004
[27] Hansen P C 1988 Regularization, gsvd and truncated gsvd (generalized singular value decomposition) *BIT Numerical Mathematics* **29** 491–504

[28] Antil H and Rautenberg C 2019 Sobolev spaces with non-muckenhoupt weights, fractional elliptic operators, and applications *SIAM J. Math. Anal.* **51** 2479–503
[29] Weiss C J, van Bloemen Waanders B G and Antil H 2020 Fractional Operators Applied to Geophysical Electromagnetics *Geophys. J. Int.* **220** 1242–59
[30] Antil H, Berry T and Harlim J 2018 Fractional diffusion maps (arXiv:1810.03952)
[31] Bueno-Orovio A, Kay D, Grau V, Rodriguez B and Burrage K 2014 Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization *J. R. Soc. Interface* **11** 20140352
[32] Antil H, Khatri R and Warma M 2019 External optimal control of nonlocal PDEs *Inverse Problems* **35** 084003
[33] Antil H, Verma D and Warma M 2020 External optimal control of fractional parabolic PDEs *ESAIM Control Optim. Calc. Var.* **26** 20
[34] Bougleux S, Elmoataz A and Melkemi M 2009 Local and nonlocal discrete regularization on weighted graphs for image and mesh processing *Int. J. Comput. Vis.* **84** 220–36
[35] Liu W, Ma X, Zhou Y, Tao D and Cheng J 2019 *p*-Laplacian regularization for scene recognition *IEEE Transactions on Cybernetics* **49** 2927–40
[36] Magiera J, Ray D, Hesthaven J S and Rohde C 2020 Constraint-aware neural networks for Riemann problems *Int. J. Comput. Vis.* **409** 109345
[37] Hintermüller M and Rautenberg C N 2017 Optimal selection of the regularization function in a weighted total variation model. Part I: Modelling and theory *J. Math. Imag. Vis.* **59** 498–514
[38] Hintermüller M, Rautenberg C N, Wu T and Langer A 2017 Optimal selection of the regularization function in a weighted total variation model. Part II: Algorithm, its analysis and numerical tests *J. Math. Imag. Vis.* **59** 515–33
[39] Starck J-L, Candès E J and Donoho D L 2002 The curvelet transform for image denoising *IEEE Trans. Image Process.* **11** 670–84
[40] Hansen P C 1994 Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems *Numer. Algorithms* **6** 1–35
[41] Ambrosio L, Fusco N and Pallara D 2000 *Functions of Bounded Variation and Free Discontinuity Problems* (New York: Oxford University Press)
[42] Bartels S and Milicevic M 2017 Alternating direction method of multipliers with variable step sizes (arXiv:1704.06069)
[43] Kinderlehrer D and Stampacchia G 1980 *An Introduction to Variational Inequalities and Their Applications* (New York: Academic)
[44] Di Z, Leyffer S and Wild S M 2016 Optimization-based approach for joint x-ray fluorescence and transmission tomographic inversion *SIAM J. Imag. Sci.* **9** 1
[45] Radon J 1986 On the determination of functions from their integral values along certain manifolds *IEEE Trans. Med. Imaging* **5** 170–6
[46] Colton D and Kress R 2013 *Inverse Acoustic and Electromagnetic Scattering Theory* (Appl. Math. Sci of vol 93) 3rd edn (New York: Springer)
[47] Nash S G 2000 A survey of truncated-Newton methods *J. of Comp. and App. Math.* **124** 45–59
[48] Hansen P C and O'Leary D P 1993 The use of the l-curve in the regularization of discrete ill-posed problems *SIAM J. Sci. Comput.* **14** 1487–503
[49] Vogel C R 1996 Non-convergence of the *L*-curve regularization parameter selection method *Inverse Problems* **12** 535–47
[50] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR*) (June 2016) pp 770–8
[51] Wu S, Zhong S and Liu Y 2018 Deep residual learning for image steganalysis *Multimed. Tool. Appl.* **77** 10437–53
[52] Chen H, Dou Q, Yu L, Qin J and Voxresnet P-A H 2018 Deep voxelwise residual networks for brain segmentation from 3d mr images *NeuroImage* **170** 446–55
[53] Lee D, Yoo J, Tak S and Ye J C 2018 Deep residual learning for accelerated mri using magnitude and phase networks *IEEE Trans. Biomed. Eng.* **65** 1985–95
[54] Bischke B, Bhardwaj P, Gautam A, Helber P, Borth D and Dengel A 2017 Detection of flooding events in social multimedia and satellite imagery using deep neural networks *Working Notes Proceedings of the MediaEval 2017* (MediaEval Benchmark, September) (Dublin: MediaEval) vol 13–15
[55] Tai Y, Yang J and Liu X 2017 Image super-resolution via deep recursive residual network *IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR*) (July 2017) pp 2790–8

[56] Zhang Q, Yuan Q, Zeng C, Li X and Wei Y 2018 Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network *IEEE Trans. Geosci. Remote Sens.* **56** 4274–88
[57] Kelley C T 1999 Iterative methods for optimization *Frontiers in Applied Mathematics* (Philadelphia: SIAM)
[58] Bottou L, Curtis F E and Nocedal J 2018 Optimization methods for large-scale machine learning *SIAM Rev.* **60** 223–311
[59] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[60] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Berlin: Springer)
[61] Higham C F and Higham D J 2019 Deep learning: an introduction for applied mathematicians *SIAM Rev.* **61** 860–91
[62] Sjöberg J 1992 Overtraining regularization, and searching for minimum in neural networks *IFAC Proceedings Volumes* **25** 73–8
[63] Antil H, Pfefferer J and Rogovs S 2018 Fractional operators with inhomogeneous boundary conditions: analysis, control, and discretization *Commun. Math. Sci.* **16** 1395–426
[64] Clarke F H 1975 Generalized gradients and applications *Trans. Amer. Math. Soc.* **205** 247–62
[65] Austin A P, Di Z, Leyffer S and Wild S M 2019 Simultaneous sensing error recovery and tomographic inversion using an optimization-based approach *SIAM J. Sci. Comput.* **41** B497–521